



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 III курсу

ФПМ групи КВ-23

Марінченко Михайло Олексійович

Перевірив: Петрашенко А.В.

Київ – 2024

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 кожен з таблиць засобами pgAdmin 4.

Опис предметної області «Платформа для бронювання та управління спортивними заняттями»

Обрана предмета область передбачає бронювання на спортивні заняття та отримання даних про заняття. Інформація про тренерів, клієнтів, заняття(час, місце проведення) зберігається в окремих таблицях.

Опис сутностей предметної області

1. Клієнт(Client), з атрибутами: код клієнта, ім'я, прізвище, електронна пошта, номер телефона, дата реєстрації. Призначена для збереження інформації про клієнта;
2. Бронювання (Bookings), з атрибутами: код бронювання, код клієнта, код сесії, дата бронювання, статус бронювання. Призначена для збереження інформації про бронювання відповідним клієнтом на відповідну сесію.
3. Сесія (Sessions), з атрибутами: код сесії, код тренера, код локації, час початку, час закінчення, максимальна кількість учасників, ціна. Призначена для збереження інформації про сесію тренування;
4. Локації (Locations), з атрибутами: код локації, назва, адреса, місто, містивність. Призначена для збереження інформації про місце проведення тренування.
5. Тренери(Instructors), з атрибутами: код тренера, досвід роботи, опис(біографія), рейтинг. Призначена для збереження інформації про тренерів.

Опис зв'язків між сутностями предметної області

Сутність “Клієнт” має зв'язок 1:N по відношенню до сутності “Бронювання” оскільки клієнт може містити декілька бронювань.

Сутність “Сесія” має зв'язок 1:N по відношенню до “Бронювань”, оскільки одна сесія може бути заброньована декілька разів.

Сутність “Локація” має зв'язок 1:N по відношенню до “Сесія”, оскільки одна локація може бути місцем проведення для багатьох сесій, але сесія має лише одне місце проведення.

Сутність “Тренер” має зв'язок 1:N по відношенню до “Сесія”, оскільки один тренер може проводити декілька тренувальних сесій, але у сесії завжди один тільки тренер.

Логічна модель (схема) БД «Платформа для бронювання та управління спортивними заняттями»

Логічну модель (схему бази даних) наведено на рисунку 1.

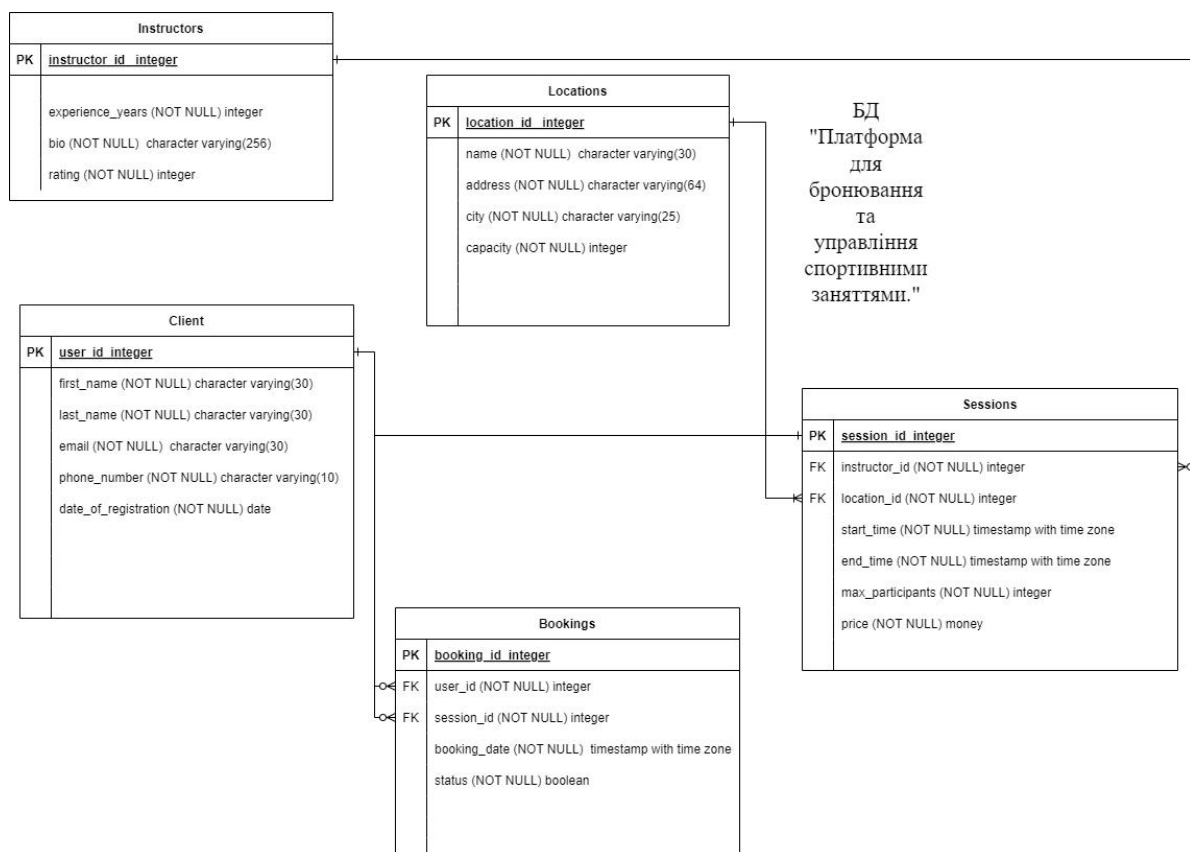


Рисунок 1 - Схема бази даних (інструмент: draw.io)

Опис об'єктів бази даних у вигляді таблиці

Клієнт		
Атрибут	Опис атрибуту	Тип даних
user_id[PK]	Унікальний ідентифікатор клієнта	integer
first_name	Ім'я користувача	Character varying(30)
last_name	Прізвище користувача	Character varying(30)
email	Електронна пошта користувача	Character varying(30)
phone_number	Мобільний номер телефону користувача	Character varying(30)
date_of_registration	Дата реєстрації	date

Таблиця 1 - Опис "Клієнт"

Бронювання		
Атрибут	Опис атрибуту	Тип даних
booking_id	Унікальний ідентифікатор бронювання	integer
user_id	Унікальний ідентифікатор клієнта	integer
session_id	Унікальний ідентифікатор сесії	integer
booking_date	Дата бронювання	Timestamp with time zone
status	Статус бронювання	boolean

Таблиця 2 - Опис "Бронювання"

Сесії		
Атрибут	Опис атрибуту	Тип даних
session_id	Унікальний ідентифікатор сесії	integer
Instructor_id	Унікальний ідентифікатор тренера	integer
location_id	Унікальний ідентифікатор локації	integer
start_time	Час початку	Timestamp with time zone
end_time	Час закінчення	Timestamp with time zone
max_participants	Максимальна кількість учасників	integer
price	Ціна	money

Таблиця 3 - Опис “Сесії”

Локація		
Атрибут	Опис атрибуту	Тип даних
location_id	Унікальний ідентифікатор локації	integer
name	Назва локації	Character varying(30)
address	Адреса локації	Character varying(64)
city	Місто, де знаходиться локація	Character varying(25)
capacity	Містивність локації(Кількість людей)	integer

Таблиця 4 - Опис “Локація”

Тренери		
Атрибут	Опис атрибуту	Тип даних
instructor_id	Унікальний ідентифікатор тренера	integer
experience_years	Досвід роботи в роках	integer
bio	Описання тренера	Character varying(256)
rating	Рейтинг тренера	integer

Таблиця 5 - Опис "Тренери"

Функціональні залежності для кожної таблиці

1. Instructors

- **instructor_id** → { experience_years, bio, rating}
 - Первинний ключ (instructor_id) унікально ідентифікує записи в цій таблиці.

2. Client

- **user_id** → {first_name, last_name, email, phone_number, date_of_registration}
 - Первинний ключ (user_id) визначає всі інші атрибути клієнта.

3. Locations

- **location_id** → {name, address, city, capacity}
 - Первинний ключ (location_id) унікально ідентифікує локації.

4. Bookings

- **booking_id** → {user_id, session_id, booking_date, status}
 - Первинний ключ (booking_id) визначає всі інші атрибути бронювання.
- **user_id** → {first_name, last_name, email, phone_number, date_of_registration}
 - Ця залежність існує через зв'язок з таблицею Client.

5. Sessions

- **session_id** → {instructor_id, location_id, start_time, end_time, max_participants, price}
 - Первинний ключ (session_id) визначає всі деталі сесії.
- **instructor_id** → {experience_years, bio, rating}
 - Ця залежність існує через зв'язок з таблицею Instructors.
- **location_id** → {name, address, city, capacity}
 - Ця залежність існує через зв'язок з таблицею Locations.

Відповідність схеми бази даних нормальним формам

Схема відповідає 1НФ, тому що в таблиці немає дубльованих рядків, у кожній комірці зберігається атомарне значення, у кожному стовпчику зберігаються дані одного типу.

Схема відповідає 2НФ, тому що вона відповідає 1НФ, має первинний ключ від якого залежать всі не ключові стовпчики таблиці.

Схема відповідає 3НФ, тому що вона відповідає 2НФ і всі не ключові атрибути таблиці залежать від первинного ключа.

Таблиці бази даних у pgAdmin4

Зображені на малюнках 2.1 - 2.5

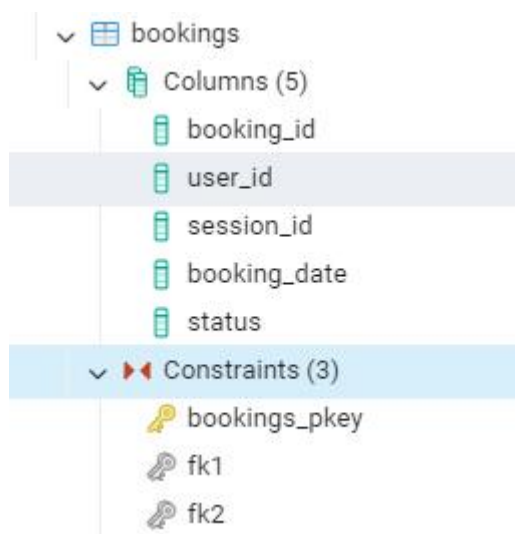


Рисунок 2.1

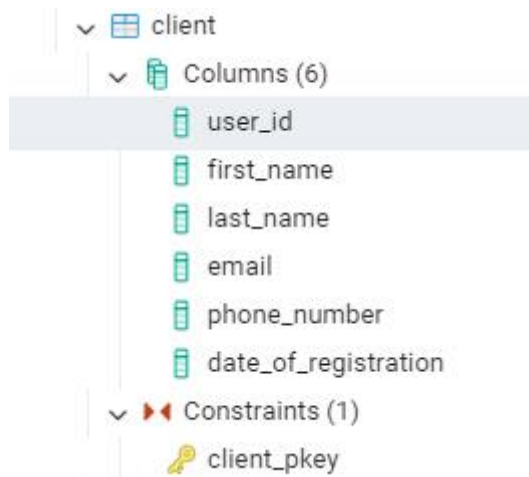


Рисунок 2.2

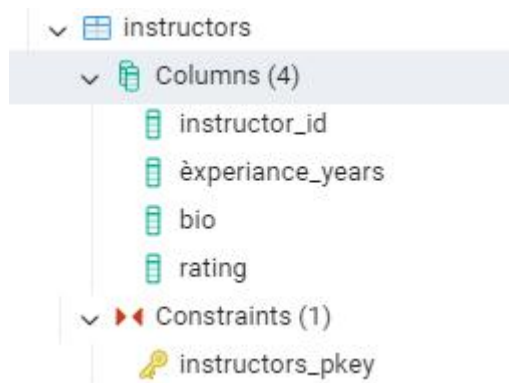


Рисунок 2.3



Рисунок 2.4

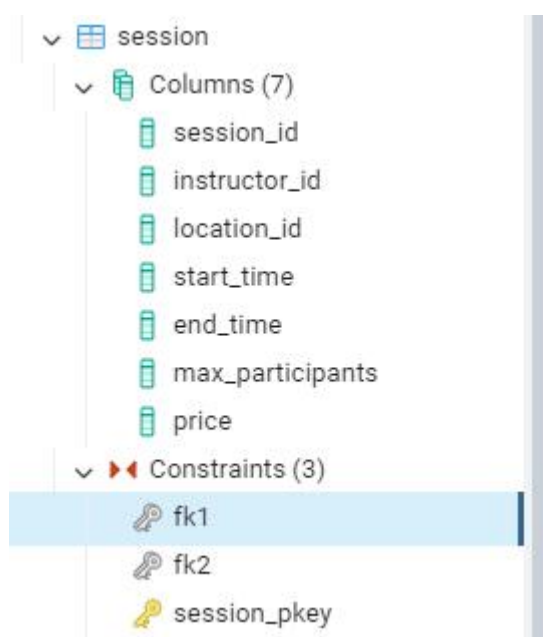


Рисунок 2.5

Фотографії вмісту таблиць

	booking_id [PK] integer	user_id integer	session_id integer	booking_date timestamp with time zone	status boolean
1	1	1	3	2024-06-10 23:00:00+03	true
2	2	2	1	2024-10-10 15:00:00+03	true

Рисунок 3.1

	user_id [PK] integer	first_name character varying (30)	last_name character varying (30)	email character varying (30)	phone_number character varying (10)	date_of_registration date
1	1	Bogdan	Peretiatko	student@gmail.com	0996540113	2024-09-24
2	2	Kirill	Grovny	student1@gmail.com	0504502312	2023-06-10

Рисунок 3.2

	instructor_id [PK] integer	experience_years integer	bio character varying (256)	rating integer
1	1	12	Very good man	7
2	2	5	New man	8

Рисунок 3.3

	location_id [PK] integer	name character varying (30)	address character varying (64)	city character varying (25)	capacity integer
1	1	Sportzal	st. Dovnar 4	Kiev	10
2	2	Gym-2	st. Beresteyski 28	Kiev	15

Рисунок 3.4

	session_id [PK] integer	instructor_id integer	location_id integer	start_time timestamp with time zone	end_time timestamp with time zone	max_participants integer	price money
1	1	1	1	2025-04-05 06:00:00+03	2025-04-05 08:00:00+03	15	10,00 ?
2	2	1	1	2025-04-05 15:00:00+03	2025-04-05 17:00:00+03	10	15,00 ?
3	3	2	2	2025-05-05 17:00:00+03	2025-05-05 21:00:00+03	15	12,00 ?
4	4	2	2	2025-04-05 06:00:00+03	2025-04-05 08:00:00+03	10	20,00 ?

Рисунок 3.5

Посилання на репозиторій: <https://github.com/GekJack/DBrepos.git>

SQL-текст опису БД

```
SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

SET default_tablespace = '';

SET default_table_access_method = heap;

CREATE TABLE public.bookings (
    booking_id integer NOT NULL,
    user_id integer NOT NULL,
    session_id integer NOT NULL,
    booking_date timestamp with time zone NOT NULL,
    status boolean NOT NULL
);
```

```
ALTER TABLE public.bookings OWNER TO postgres;
```

```
CREATE SEQUENCE public.bookings_booking_id_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;
```

```
ALTER SEQUENCE public.bookings_booking_id_seq OWNER TO postgres;
```

```
ALTER SEQUENCE public.bookings_booking_id_seq OWNED BY  
public.bookings.booking_id;
```

```
CREATE TABLE public.client (  
    user_id integer NOT NULL,  
    first_name character varying(30) NOT NULL,  
    last_name character varying(30) NOT NULL,  
    email character varying(30) NOT NULL,  
    phone_number character varying(10) NOT NULL,
```

```
    date_of_registration date NOT NULL  
);
```

```
ALTER TABLE public.client OWNER TO postgres;
```

```
CREATE TABLE public.instructors (  
    instructor_id integer NOT NULL,  
    "experience_years" integer NOT NULL,  
    bio character varying(256) NOT NULL,  
    rating integer NOT NULL  
);
```

```
ALTER TABLE public.instructors OWNER TO postgres;
```

```
CREATE TABLE public.locations (  
    location_id integer NOT NULL,  
    name character varying(30) NOT NULL,  
    address character varying(64) NOT NULL,  
    city character varying(25) NOT NULL,  
    capacity integer NOT NULL  
);
```

```
ALTER TABLE public.locations OWNER TO postgres;
```

```
CREATE TABLE public.session (  
    session_id integer NOT NULL,  
    instructor_id integer NOT NULL,  
    location_id integer NOT NULL,  
    start_time timestamp with time zone NOT NULL,  
    end_time timestamp with time zone NOT NULL,  
    max_participants integer NOT NULL,  
    price money NOT NULL  
);
```

```
ALTER TABLE public.session OWNER TO postgres;
```

```
CREATE SEQUENCE public.session_session_id_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;
```

```
ALTER SEQUENCE public.session_session_id_seq OWNER TO postgres;
```

```
ALTER SEQUENCE public.session_session_id_seq OWNED BY public.session.session_id;
```

```
ALTER TABLE ONLY public.bookings ALTER COLUMN booking_id SET DEFAULT  
nextval('public.bookings_booking_id_seq'::regclass);
```

```
ALTER TABLE ONLY public.session ALTER COLUMN session_id SET DEFAULT  
nextval('public.session_session_id_seq'::regclass);
```

```
ALTER TABLE ONLY public.bookings
```

```
ADD CONSTRAINT bookings_pkey PRIMARY KEY (booking_id);
```

```
ALTER TABLE ONLY public.client
```

```
ADD CONSTRAINT client_pkey PRIMARY KEY (user_id);
```



```
ALTER TABLE ONLY public.instructors
```

```
ADD CONSTRAINT instructors_pkey PRIMARY KEY (instructor_id);
```

```
ALTER TABLE ONLY public.locations
```

```
ADD CONSTRAINT locations_pkey PRIMARY KEY (location_id);
```

```
ALTER TABLE ONLY public.session
```

```
ADD CONSTRAINT session_pkey PRIMARY KEY (session_id);
```

```
ALTER TABLE ONLY public.bookings
```

```
ADD CONSTRAINT fk1 FOREIGN KEY (user_id) REFERENCES  
public.client(user_id);
```

```
ALTER TABLE ONLY public.session
```

```
ADD CONSTRAINT fk1 FOREIGN KEY (instructor_id) REFERENCES  
public.instructors(instructor_id);
```

```
ALTER TABLE ONLY public.bookings  
    ADD CONSTRAINT fk2 FOREIGN KEY (session_id) REFERENCES  
public.session(session_id);
```

```
ALTER TABLE ONLY public.session  
    ADD CONSTRAINT fk2 FOREIGN KEY (location_id) REFERENCES  
public.locations(location_id);
```