МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет

«Харківський політехнічний інститут»

Кафедра «Стратегічного управління»

ЗВІТ

З лабораторної роботи № 2

На тему «ЛОГІЧНЕ ПРОЕКТУВАННЯ БАЗИ ДАНИХ»

з курсу «Програмування баз даних та знань»

Виконавши студент групи КН-м524

Баранецький І.Б.

Перевірив викладач

Лисенко О.О.

Харків 2024

Зміст

[1. Завдання: 4](#_Toc177993983)

[2. Виконання: 4](#_Toc177993984)

[2.1 Створити схему бази даних. 4](#_Toc177993985)

[2.2 Заповнення даних. 4](#_Toc177993986)

[2.3 Забір даних. 5](#_Toc177993987)

[2.4 Застосування. 5](#_Toc177993988)

[3. Висновки: 7](#_Toc177993989)

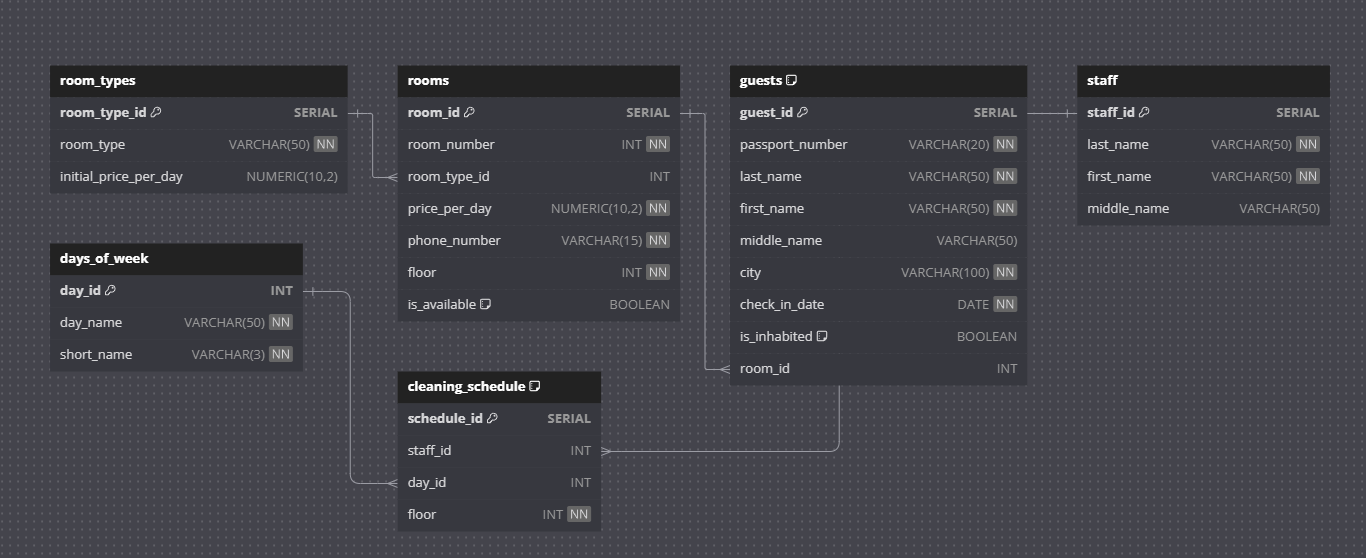
[Додаток А 8](#_Toc177993990)

1. Завдання:

Виконати етап логічного проектування реляційної бази даних та створити спроектовану базу даних засобами обраної СУБД.

1. Виконання:
   1. Створити схему бази даних.

На основі діаграми «Сутність-зв’язок» з першої лабораторної роботи малюнок 1:



Малюнок 1 - Діаграма сутність-зв'язок предметної області у нотації Мартіна

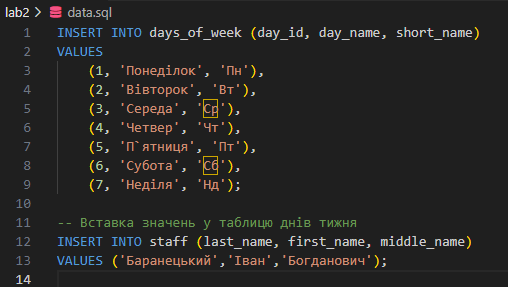
Було створено 6 таблиць для реалізації предметної області. Скрипт таблиць зазначено в додатку А.

* 1. Заповнення даних.

Для деяких таблиць я додав скрипти заповнення даних деяких таблиць. Така таблиця як «Дні Тижня» мають статичний набір даних, тому для неї було створено простий скрипт заповнення значеннями, який додає всі дні тижня зі скороченими назвами. Це дозволяє уникнути дублювання даних та спрощує управління інформацією. Ця таблиця може заповнюватись даним, наприклад для додавання перекладу на інші мови .

Також було додано скрипт для вставки даних у таблицю службовців (таблиця «staff»), що містить базову інформацію про співробітників: прізвище, ім’я та по батькові.

Ці скрипти зазначені на малюнку 2:



Малюнок 2 – Скрипт вставки даних

* 1. Забір даних.

Для перевірки коректності заповнення бази даних та роботи системи було виконано кілька запитів для забору даних. Запит на отримання всіх вільних номерів у готелі, малюнок 3:



Малюнок 3 – Запит на отримання вільних кімнат

Даний запит дозволяє отримати інформацію про всі доступні номери, які готові для заселення. Це корисно для адміністратора готелю, який шукає вільні номери для клієнтів. Запит на отримання інформації про службовців готелю, зображений на малюнку 4 :



Малюнок 3 – Запит на отримання інформації про службовців готелю

Цей запит повертає прізвища та імена всіх співробітників, які працюють у готелі. Запит можна використовувати для перегляду базової інформації про службовців, що відповідають за прибирання номерів.

Такі запити дозволяють ефективно отримувати потрібні дані з бази для подальшого використання в системі управління готелем.

* 1. Застосування.

Щоб отримати доступ до створеної бази даних і виконати запити, необхідно виконати деякі кроки.

* Спочатку треба завантажити та встановити Docker на свою

операційну систему.

* Після того, як Docker буде встановлений, відкрийте термінал будь-

яким зручним способом.

* Треба перейти до директорії виконання лабораторної роботи за

допомогою команди «*cd».*

* У папці зі скриптами виконайте команду для запуску Docker-

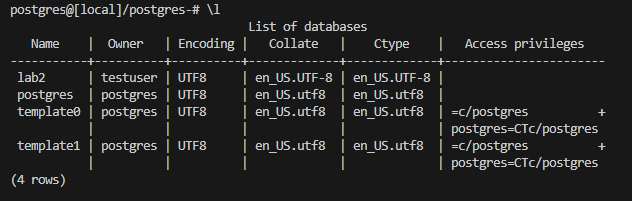
контейнера: «.\\_create.bat». Ця команда створить контейнер з PostgreSQL та підключить SQL-скрипти для ініціалізації бази даних.

* Після запуску контейнера для доступу до бази даних виконайте

команду «.\\_psql.bat». Ця команда відкриє інтерфейс PostgreSQL всередині контейнера.

* Після підключення введіть команду для перевірки існуючих баз

даних «\l». Результат вказаний на малюнку 4:



Малюнок 4 – Список баз даних

* Щоб підключитися до конкретної бази треба ввести команду «\c

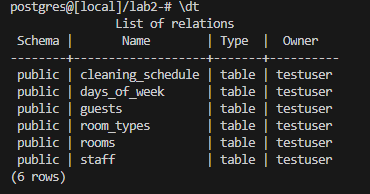
<Назва бази>». Результат вказаний на малюнку 5:



Малюнок 5 – Підключення до конкретної бази

* Перевірити наявні таблиці можна за допомогою команди «\dt».

Результат вказаний на малюнку 6:



Малюнок 6 - Виведення таблиць у вибраній базі

1. Висновки:

У цій лабораторній роботі було виконано етап логічного проектування реляційної бази даних, що включає розробку схеми таблиць, визначення зв’язків між ними та створення бази даних засобами обраної системи управління базами даних (СУБД) PostgreSQL. В ході роботи було створено кілька таблиць, що відображають сутності предметної області. Було визначено основні атрибути, первинні та зовнішні ключі, а також додано індекси для оптимізації запитів. Крім того, розроблено скрипти для заповнення таблиць тестовими даними та отримання інформації з бази. Окрім цього, було налаштовано Docker-контейнер для середовища бази даних. Завдяки цьому підходу вдалося спростити процес ініціалізації та тестування бази даних.

Додаток А

-- Таблиця для типів номерів

CREATE TABLE room\_types (

    room\_type\_id SERIAL PRIMARY KEY,

    room\_type VARCHAR(50) NOT NULL,

    initial\_price\_per\_day NUMERIC(10, 2)

);

-- Таблиця для номерів у готелі

CREATE TABLE rooms (

    room\_id SERIAL PRIMARY KEY,

    room\_number INT NOT NULL,

    room\_type\_id INT REFERENCES room\_types(room\_type\_id),

    price\_per\_day NUMERIC(10, 2) NOT NULL,

    phone\_number VARCHAR(15) NOT NULL,

    floor INT NOT NULL,

    is\_available BOOLEAN DEFAULT true

);

-- Таблиця для клієнтів

CREATE TABLE guests (

    guest\_id SERIAL PRIMARY KEY,

    passport\_number VARCHAR(20) NOT NULL,

    last\_name VARCHAR(50) NOT NULL,

    first\_name VARCHAR(50) NOT NULL,

    middle\_name VARCHAR(50),

    city VARCHAR(100) NOT NULL,

    check\_in\_date DATE NOT NULL,

    is\_inhabited BOOLEAN DEFAULT false,

    room\_id INT REFERENCES rooms(room\_id)

);

CREATE INDEX idx\_passport\_number ON guests(passport\_number);

-- Таблиця для службовців

CREATE TABLE staff (

    staff\_id SERIAL PRIMARY KEY,

    last\_name VARCHAR(50) NOT NULL,

    first\_name VARCHAR(50) NOT NULL,

    middle\_name VARCHAR(50)

);

-- Таблиця для днів тижня

CREATE TABLE days\_of\_week (

    day\_id INT PRIMARY KEY,

    day\_name VARCHAR(50) NOT NULL,

    short\_name VARCHAR(3) NOT NULL

);

-- Таблиця для графіку прибирання

CREATE TABLE cleaning\_schedule (

    schedule\_id SERIAL PRIMARY KEY,

    staff\_id INT REFERENCES staff(staff\_id),

    day\_id INT REFERENCES days\_of\_week(day\_id),

    floor INT NOT NULL,

    UNIQUE (staff\_id, day\_id, floor)

);