

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

ФПМ групи КВ-23

Атанов Назар Данилович

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 кожну з таблиць засобами pgAdmin 4.

**Опис предметної області «Онлайн-магазин для продажу електроніки»**

Обрана предметна область передбачає платформу з бронювання та організації подорожей

**Опис сутностей предметної області**

1. Користувач(user) з атрибутами: код користувача, ім’я. Призначена для зберігання інформації про користувача

2. Подорож (trip) з атрибутами: код подорожі, назва, дата початку, дата кінця. Сутність призначена для вибору подорожі і отримання інформації про неї

3. Бронювання(booking) з атрибутами: код бронювання, статус, дата початку, дата кінця. Зберігає інформацію про бронювання користувачем в певній подорожі

4. Проживання(accommodation) з атрибутами: код проживання, адреса, тип, назва. Ця сутність зберігає деталі про проживання, пов’язані з подорожжю. Вона дозволяє зв’язувати бронювання з конкретними місцями проживання.

**Опис зв’язків між сутностями предметної області**

Сутність Користувач має зв’язок 1:N з сутністю подорож, оскільки користувач може мати декільки подорожей

Сутність Подорож має зв’язок 1:N з бронюванням, так як одна подорож може мати декілька бронювань

Сутність Бронювання має зв’язок 1:1 з місцем проживання, оскільки кожне бронювання прив’язане до одного міся проживання

**Концептуальна модель предметної області**

Концептуальна модель наведена на рисунку 1.

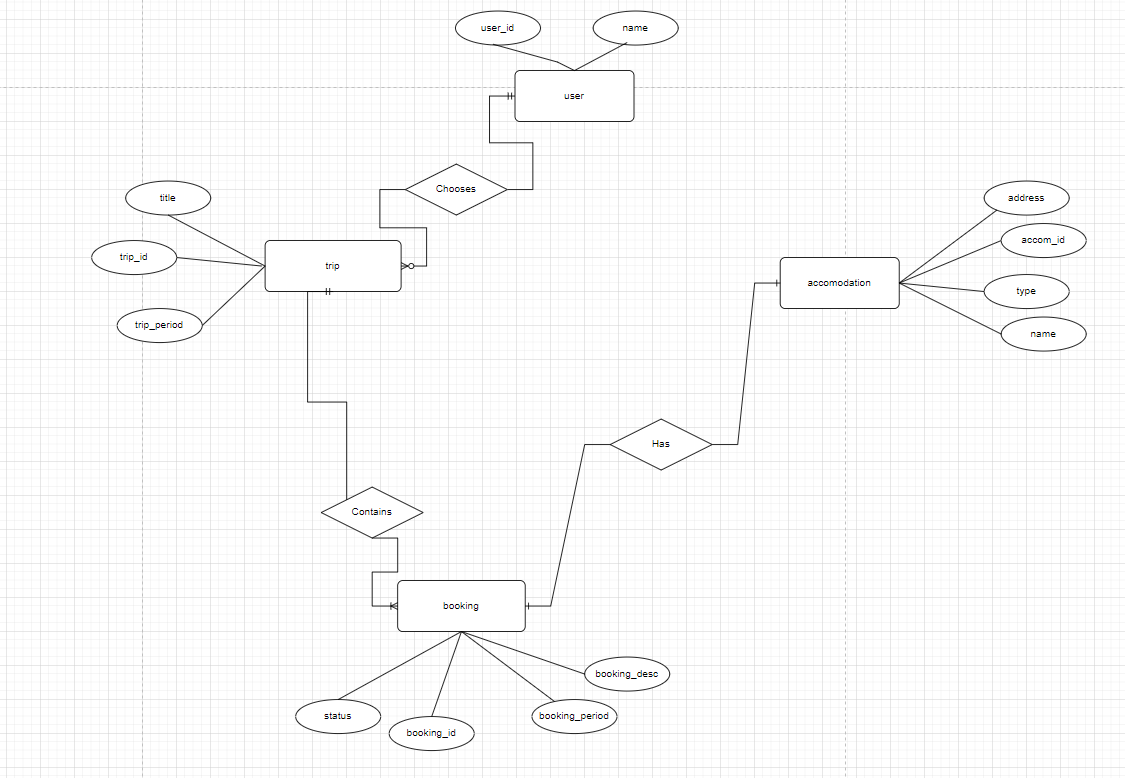


Рисунок 1 - ER-діаграма

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

Сутність “Користувач” перетворено у таблицю “user”.

Сутність “Подорож” перетворено у таблицю “trip”.

Сутність “Бронювання” перетворено у таблицю “booking”.

Сутність “Проживання” перетворено у таблицю “accommodation”.

**Логічна модель (схема) БД «Магазин»**

Логічну модель (схему бази даних) наведено на рисунку 2.

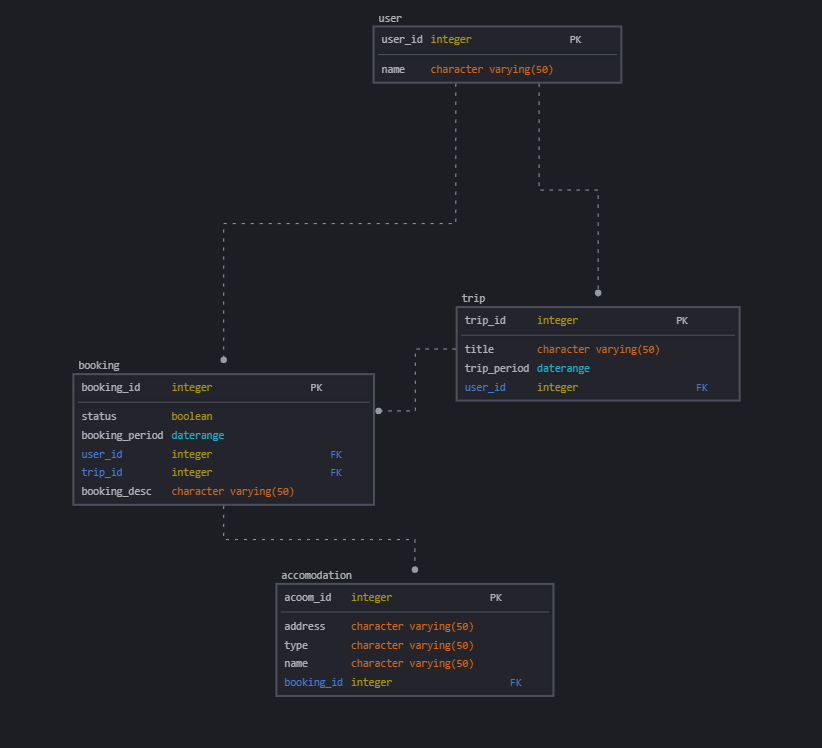


Рисунок 2 - Схема бази даних (інструмент: sqldbm.com)

**Опис об’єктів бази даних у вигляді таблиці**

Таблиця 1 - Опис структури бази даних “Магазин”

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| **user** – містить всіх користувачів | **user\_id –** унікальний ідентифікатор користувача  **name –** ім’я користувача  Не допускає NULL | **integer** (числовий)  **character varying** (рядок) |
| **trip** – містить інформацію подоріж | **trip\_id –** унікальний ідентифікатор подорожі  **title –** назву подорожі. Не допускає NULL  **user\_id -** унікальний ідентифікатор користувача  **trip\_period -** тривалість подорожі | **integer** (числовий)  **character varying** (рядок)  **integer** (числовий)  **daterange**(діапазон дат) |
| **booking –** містить інформацію про бронювання | **booking\_id –** унікальний ідентифікатор бронювання  **trip\_id –** унікальний ідентифікатор подорожі  **user\_id –** унікальний ідентифікатор користувача  **status –** статус бронювання  **booking\_period –** тривалість бронювання  **booking\_desc –** опис бронювання | **integer** (числовий)  **character varying** (рядок)  **integer** (числовий)  **boolean**  **daterange**(діапазон дат)  **character varying** (рядок) |
| **accommodation –** містить інформацію про місце проживання | **accom\_id** – унікальний ідентифікатор місця проживання  **booking\_id –** унікальний ідентифікатор бронювання  **address -** адреса м.п  **type –** тип м.п  **name –** назва м.п | **integer** (числовий)  **integer** (числовий)  **character varying** (рядок)  **character varying** (рядок)  **character varying** (рядок) |

В майбутньому можливе використання типу enum замість character varying в атрибуті type, через те що типи м.п є передбачувані

**Функціональні залежності для кожної таблиці**

TRIP:

trip\_id → title (назва залежить від коду подорожі)

BOOKING:

booking\_id → booking\_desc (ім’я залежить від бронювання)

ACCOMMODATION:

accom\_id → name (назва залежить від коду м.п)

acoom\_id → address(адреса залежить від коду м.п)

accom\_id → type (тип залежить від коду м.п)

**Відповідність схеми бази даних нормальним формам:**

Схема відповідає першій нормальній формі (1НФ), оскільки всі рядки унікальні, кожна комірка містить одне неподільне значення, а дані в кожному стовпчику є однорідними за типом.

Схема задовольняє другу нормальну форму (2НФ), оскільки вона вже в 1НФ, має первинний ключ, від якого залежать усі неключові атрибути таблиці.

Схема відповідає третій нормальній формі (3НФ), оскільки вона вже в 2НФ і всі неключові атрибути залежать безпосередньо від первинного ключа, без транзитивних залежностей.

**Таблиці бази даних у pgAdmin4**

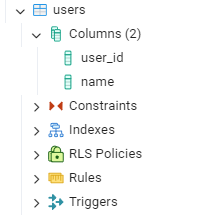
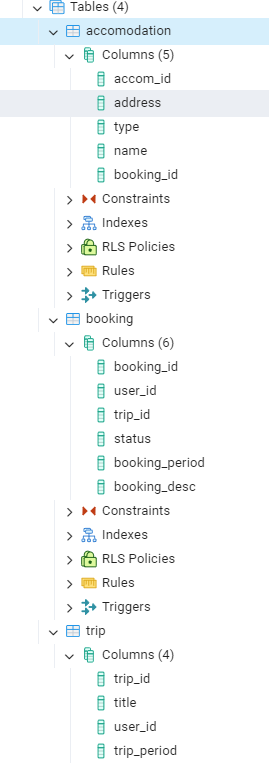


Рисунок 3 - Схема бази даних у pgAdmin 4

**Фотографії вмісту таблиць**

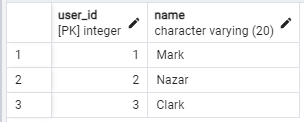


Рисунок 4.1 – вміст таблиці user

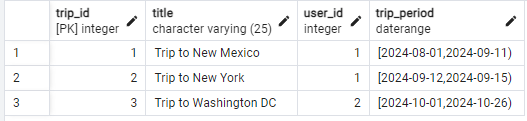


Рисунок 4.2 – вміст таблиці trip

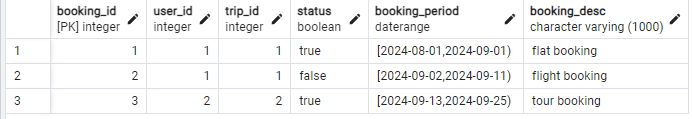


Рисунок 4.3 – вміст таблиці category

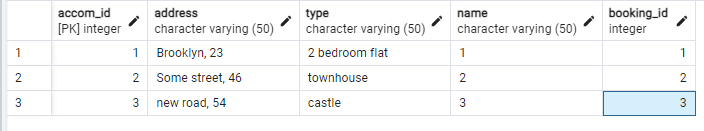


Рисунок 4.4 – вміст таблиці accommodation

Посилання на репозиторій: https://github.com/NazarBello/BD\_labs

**SQL-текст опису БД**

-- This script was generated by the ERD tool in pgAdmin 4.

-- Please log an issue at https://github.com/pgadmin-org/pgadmin4/issues/new/choose if you find any bugs, including reproduction steps.

BEGIN;

CREATE TABLE IF NOT EXISTS public.accomodation

(

accom\_id integer NOT NULL,

address character varying(50) COLLATE pg\_catalog."default" NOT NULL,

type character varying(50) COLLATE pg\_catalog."default" NOT NULL,

name character varying(50) COLLATE pg\_catalog."default" NOT NULL,

booking\_id integer NOT NULL,

CONSTRAINT accomodation\_pkey PRIMARY KEY (accom\_id)

);

CREATE TABLE IF NOT EXISTS public.booking

(

booking\_id integer NOT NULL,

user\_id integer NOT NULL,

trip\_id integer NOT NULL,

status boolean NOT NULL,

booking\_period daterange NOT NULL,

booking\_desc character varying(1000) COLLATE pg\_catalog."default",

CONSTRAINT booking\_pkey PRIMARY KEY (booking\_id)

);

CREATE TABLE IF NOT EXISTS public.trip

(

trip\_id integer NOT NULL,

title character varying(25) COLLATE pg\_catalog."default" NOT NULL,

user\_id integer NOT NULL,

trip\_period daterange NOT NULL,

CONSTRAINT trip\_pkey PRIMARY KEY (trip\_id)

);

CREATE TABLE IF NOT EXISTS public.users

(

user\_id integer NOT NULL,

name character varying(20) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT users\_pkey PRIMARY KEY (user\_id)

);

ALTER TABLE IF EXISTS public.accomodation

ADD CONSTRAINT afb FOREIGN KEY (booking\_id)

REFERENCES public.booking (booking\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.booking

ADD CONSTRAINT cwt FOREIGN KEY (trip\_id)

REFERENCES public.trip (trip\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.booking

ADD CONSTRAINT cwu FOREIGN KEY (user\_id)

REFERENCES public.users (user\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.trip

ADD CONSTRAINT t4u FOREIGN KEY (user\_id)

REFERENCES public.users (user\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

END;

**Зноска:**

Сутність booking є абстрактною сутністю від якої в подальшому можуть випливати ще більше сутностей по типу accommodation, також можливі покращення з типами данних в accommodation.type booking.booking\_desc, покращення в зміні на тип enum, це можливо за допомогою створення констант в бд, бд має гарні можливості для подальшого скейлінгу та оптимізації загалом