**Національний технічний університет України**

**“Київський політехнічний інститут імені Ігоря Сікорського”**

**Факультет прикладної математики**

**Кафедра системного програмування і спеціалізованих комп’ютерних систем**

**ЛАБОРАТОРНА РОБОТА №1**

*з дисципліни*

**“Бази даних та засоби управління”**

**ТЕМА: «Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL»**

**ПРЕДМЕТНА ГАЛУЗЬ: Платформа для бронювання та організації подорожей.**

Виконала: Сапожко В.Ю.

Студентка групи КВ-31

Перевірив(ла):

**Київ 2025**

**Мета роботи**

Здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

**Постановка задачі**

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.

**Вимоги до роботи**

1. Сутності моделі предметної галузі мають містити зв’язки типу 1:N або N:M.
2. Кількість сутностей у моделі – 3-4. Кількість атрибутів у кожній сутності: від двох до п’яти.
3. Передбачити наявність зв’язку з атрибутом.
4. Для побудови ER-діаграм використовувати одну із нотацій: Чена, “Пташиної лапки (Crow’s foot)”, UML.

**Виконання роботи  
Пункт №1 завдання**

* перелік сутностей з описом їх призначення;
* графічний файл розробленої моделі «сутність-зв’язок»;
* назва нотації.

**СУТНОСТІ:**

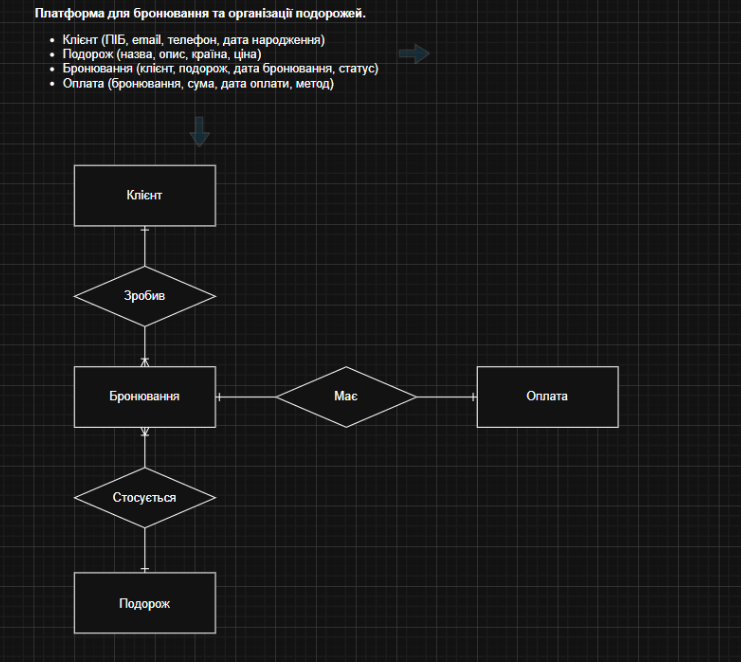
**Client (Клієнт)** - зберігає персональні дані користувача, який здійснює бронювання: ПІБ (ім’я по батькові прізвище), email, телефон, дата народження. Використовується для ідентифікації та комунікації.

**Trip (Подорож)** - довідник турів/пропозицій: назва, опис, країна (код), ціна. Використовується під час вибору та розрахунку вартості.

**Booking (Бронювання)** - факт бронювання: хто (клієнт), яку подорож, дата бронювання, статус (pending/confirmed/cancelled).

**Payment (Оплата)** - факт оплати за конкретне бронювання: сума, метод (card/cash), знижка. Прив’язаний 1:1 до бронювання (одна бронь - один платіжний запис у цій простій моделі).

**ГРАФІЧНИЙ ВИГЛЯД:**



**НОТАЦІЯ**

Crow’s Foot (Пташина лапка)

**Пункт №2 завдання**

* опис процесу перетворення (наприклад, “сутність А було перетворено у таблицю А, а зв’язок R (M:N) зумовив появу додаткової таблиці R1 тощо);
* схему бази даних у графічному вигляді з назвами таблиць (!) та зв’язками між ними, а також  необхідно намалювати перетворену ER-діаграму у ТАБЛИЦІ БД! Це означає, що тут не може бути зв'язку N:M, мають бути позначені первинні та зовнішні ключі, обмеження NOT NULL та UNIQUE і внести типи даних атрибутів.

**ОПИС ПРОЦЕСУ ПЕРЕТВОРЕННЯ:**

**Сутність "Client (Клієнт)"** була перетворена у таблицю ***client****.*  
Атрибути сутності стали стовпцями таблиці:

id - **первинний ключ** (BIGSERIAL, NOT NULL)

full\_name (VARCHAR(128), NOT NULL)

email (VARCHAR(128))

phone\_number (VARCHAR(32))

date\_of\_birth (DATE)

**Сутність "Trip (Подорож)"** була перетворена у таблицю ***trip****.*  
Атрибути сутності стали стовпцями таблиці:

id - **первинний ключ** (INTEGER, NOT NULL)

title (TEXT, NOT NULL)

description (TEXT)

country\_code (VARCHAR(3))

price (NUMERIC(12,2), NOT NULL)

**Сутність "Booking (Бронювання)"** була перетворена у таблицю ***booking****.*  
Оскільки між сутністю **Client** і сутністю **Trip** існують зв’язки **1:N**, то у таблиці booking додано зовнішні ключі.  
Атрибути таблиці:

id - **первинний ключ** (BIGSERIAL, NOT NULL)

client\_id - зовнішній ключ (INTEGER, NOT NULL)-> [client.id](http://client.id)

trip\_id - зовнішній ключ (INTEGER, NOT NULL) -> trip.id

date (DATE, NOT NULL)

status (VARCHAR(128), NOT NULL)

**Сутність "Payment (Оплата)"** була перетворена у таблицю ***payment****.*  
Оскільки між сутністю **Booking** та **Payment** є зв’язок **1:1**, ключем у таблиці payment є поле booking\_id, яке водночас виступає зовнішнім ключем на booking(id).

Атрибути таблиці:

booking\_id - зовнішній ключ (INTEGER, UNIQUE, NOT NULL)

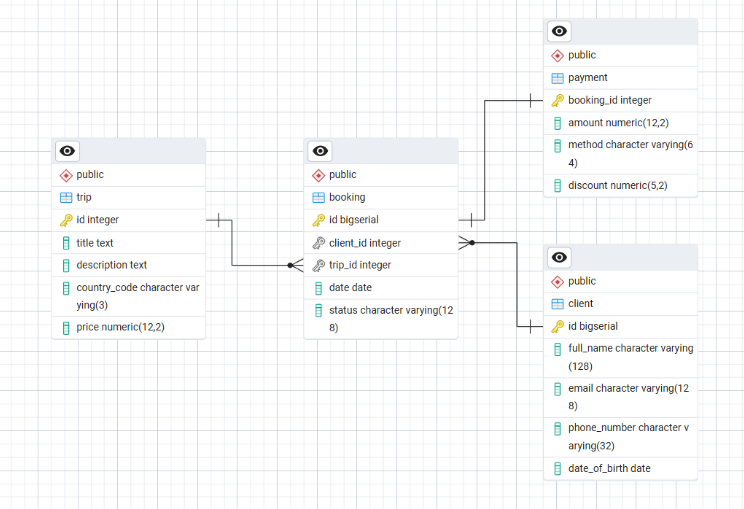
->bookin.id

amount (NUMERIC(12,2), NOT NULL)

method (VARCHAR(64), NOT NULL)

discount (NUMERIC(5,2))

**ГРАФІЧНИЙ ВИГЛЯД:**



Кожна сутність стала окремою таблицею.

Зв’язки 1:N реалізовані через зовнішні ключі у таблиці booking.

Зв’язок 1:1 реалізований через зовнішній ключ booking\_id у таблиці payment.

Зв’язків M:N у моделі немає, тому додаткових таблиць-розв’язок не утворювалося.

**Пункт №3 завдання**

* пояснення (обґрунтування!) щодо відповідності схеми бази даних нормальним формам НФ1, НФ2 та НФ3. Пояснення *полягає у наведенні функціональних залежностей*, що демонструють висновки. У випадку невідповідності надати опис необхідних змін у схемі;
* У випадку проведення змін у схемі бази даних надати оновлену версію схеми, інакше - не наводити схему.

**ФУНКЦІОНАЛЬНІ ЗАЛЕЖНОСТІ:**

**Ф3:**

trip.id -> {title, description, country\_code, price}

client.id -> {full\_name, email, phone\_number, date\_of\_birth}

booking.id -> {client\_id, trip\_id, date, status}

payment.booking\_id -> {amount, method, discount}

**НФ1:**

усі атрибути є **атомарними**;

відсутні повторювані групи атрибутів;

кожна таблиця має **первинний ключ**, що гарантує унікальність рядків.

**НФ2:**

вже відповідає умовам НФ1;

усі первинні ключі прості (складаються лише з одного атрибута: id або booking\_id), **тому часткові залежності виключені автоматично**;

усі неключові атрибути повністю залежать від первинного ключа.

**НФ3:**

відповідає вимогам НФ2;

відсутні транзитивні залежності між неключовими атрибутами (коли один неключовий атрибут визначає інший).

**АНАЛІЗ ПО ТАБЛИЦЯХ:**

У таблиці trip усі атрибути (title, description, country\_code, price) залежать лише від id. Жоден з них не визначає інший.

У таблиці client атрибути (full\_name, email, phone\_number, date\_of\_birth) залежать лише від id, і між собою незалежні.

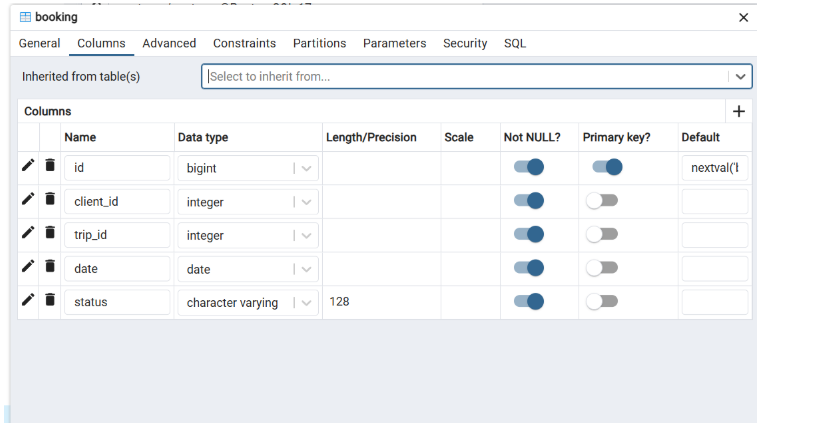
У таблиці booking атрибути (client\_id, trip\_id, date, status) залежать тільки від id. Немає ситуації, щоб, наприклад, client\_id визначав status.

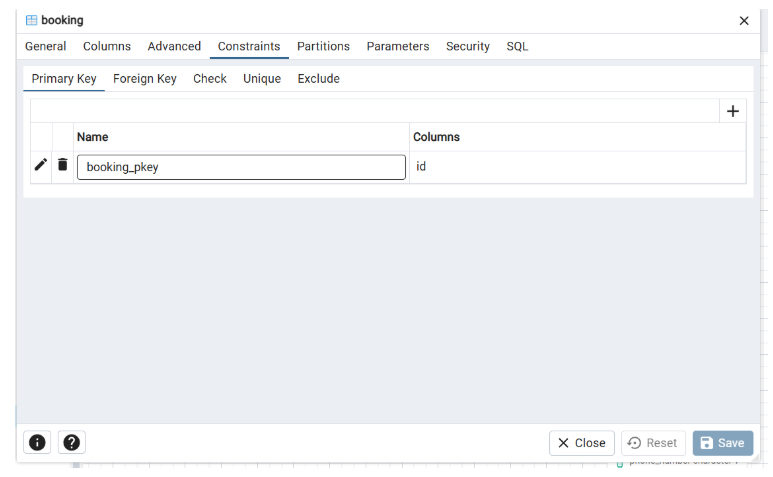
У таблиці payment атрибути (amount, method, discount) залежать лише від booking\_id. Залежностей типу method -> discount немає.

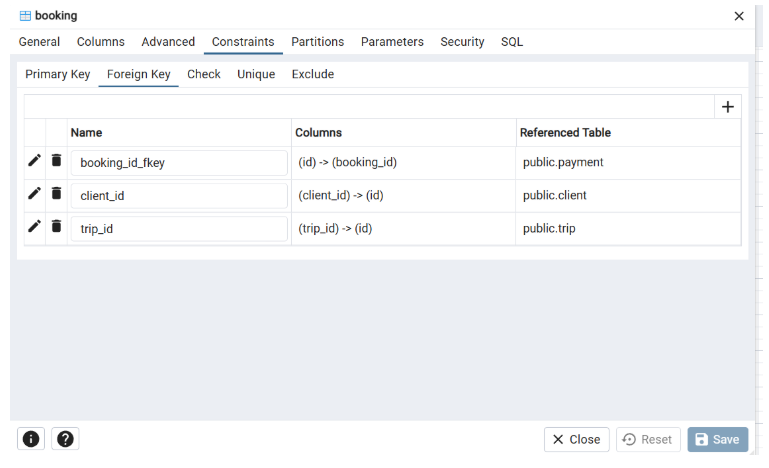
**Пункт №4 завдання**

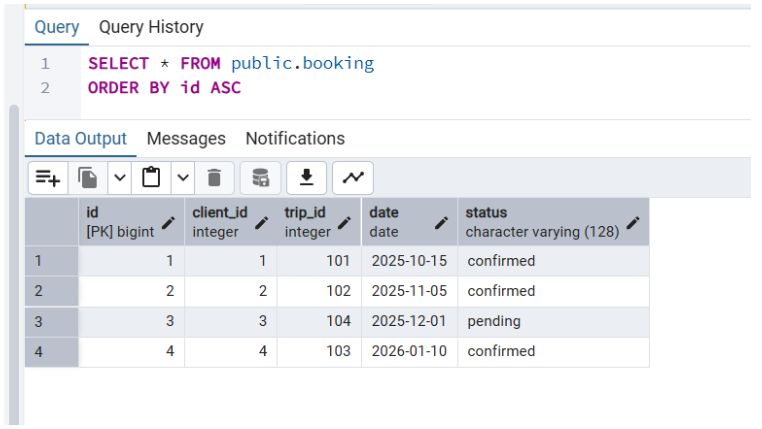
* навести копії екрану з pgAdmin4, що відображають назви, типи та обмеження на стовпці (доступне у закладці “Columns” та “Constraints” властивостей “Properties” таблиць дерева об’єктів у pgAdmin4);
* навести копії екрану з pgAdmin4, що відображають вміст таблиць бази даних у PostgreSQL. Таблиці на зображенні обов'язково повинні мати назву!

**ТАБЛИЦЯ BOOKING:**

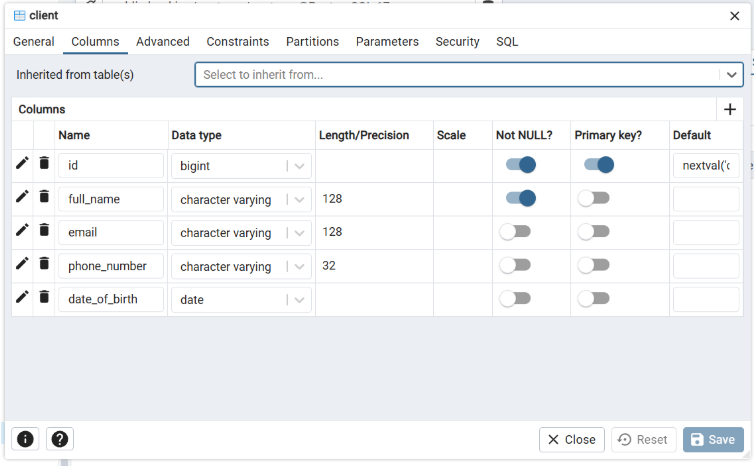
****

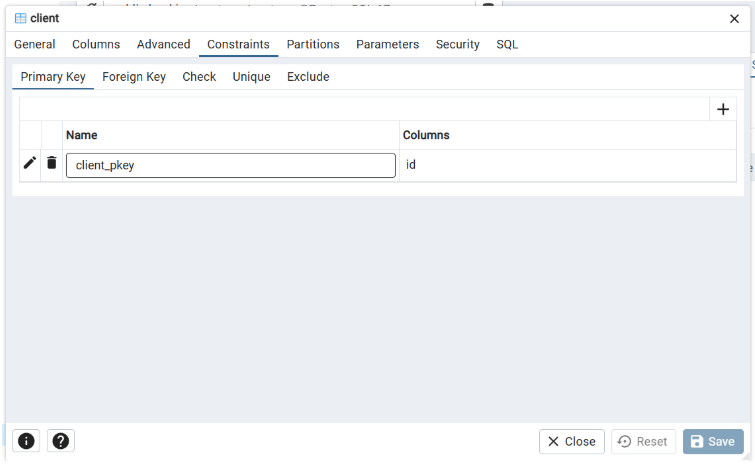
****

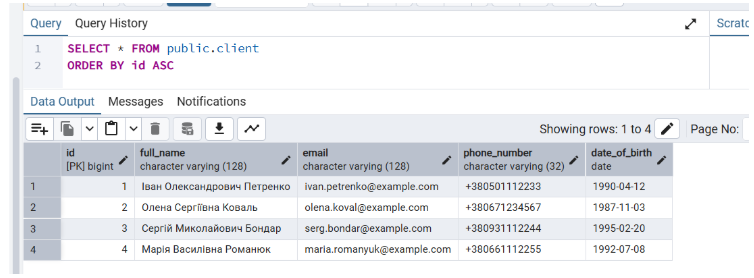
****

****

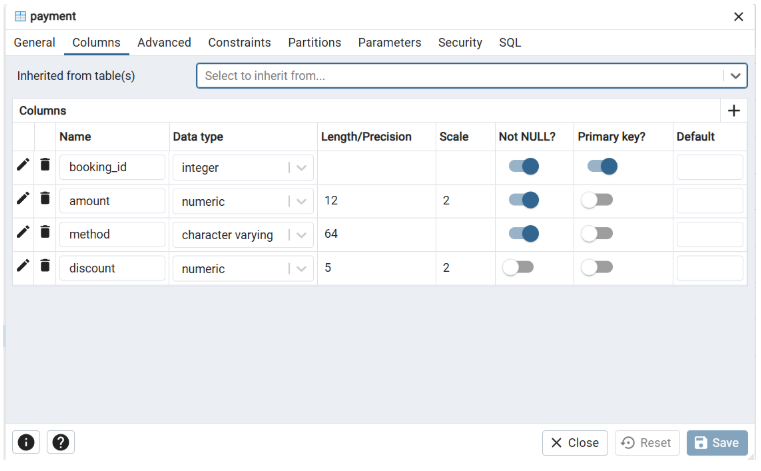
**ТАБЛИЦЯ CLIENT:**

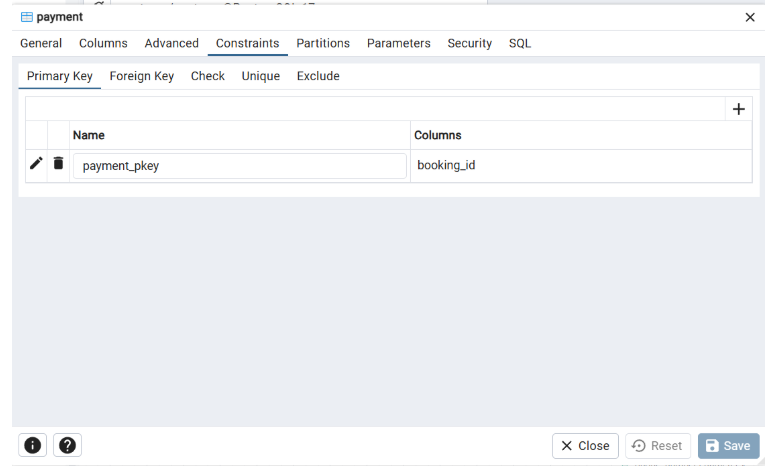
****

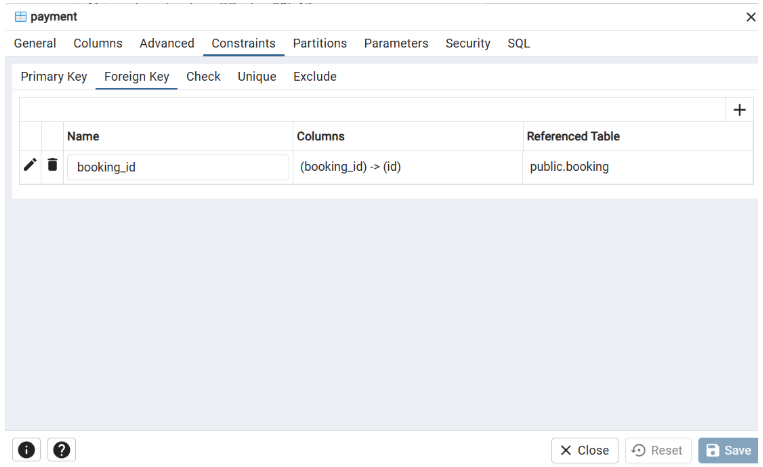
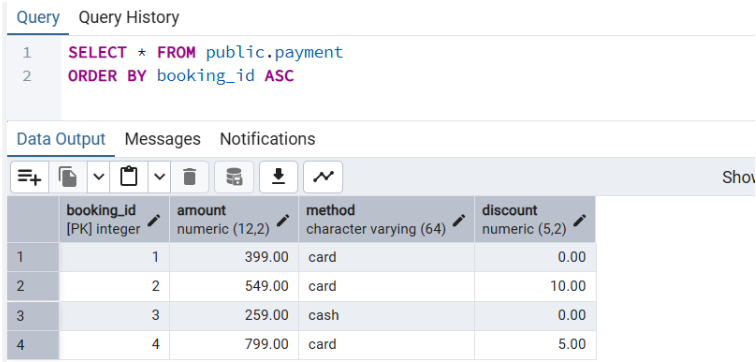
****

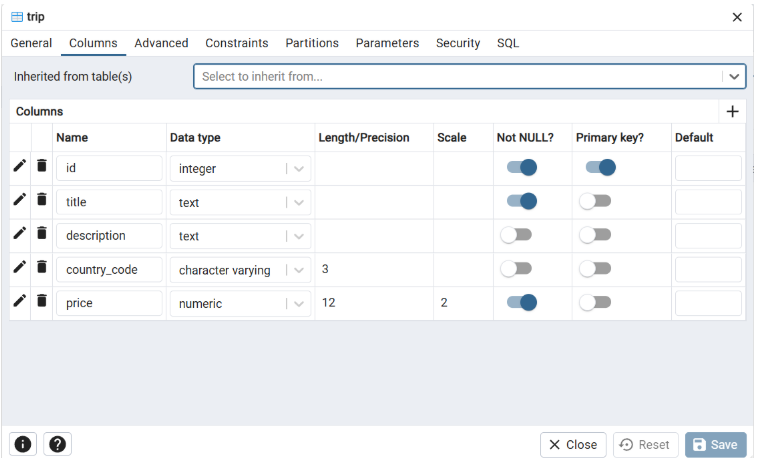
****

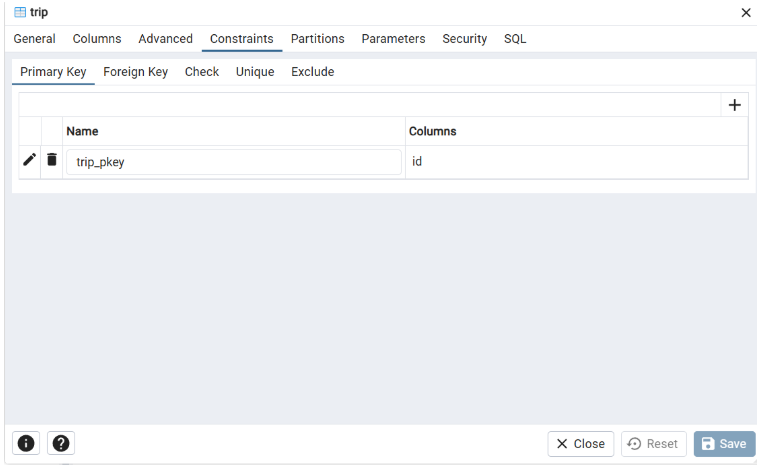
**ТАБЛИЦЯ TRIP:**

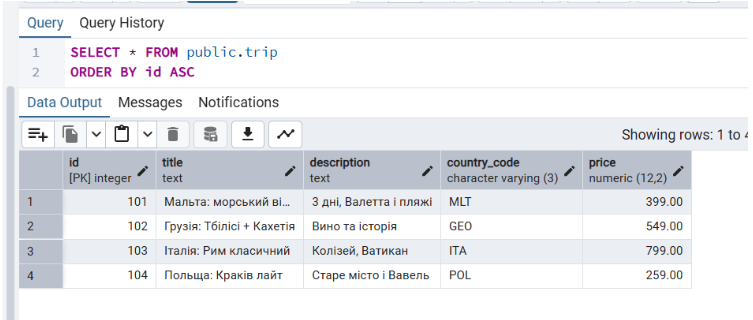
****

****







**Висновок**

У результаті виконання роботи було створено базу даних у PostgreSQL відповідно до поставлених вимог. Розроблена схема відповідає нормальним формам НФ1, НФ2 та НФ3, що підтверджується аналізом функціональних залежностей між атрибутами. Усі таблиці мають чітко визначені первинні та зовнішні ключі, атрибути є атомарними, відсутні часткові та транзитивні залежності. Це забезпечує цілісність, несуперечливість і логічну структурованість даних.

Структуру таблиць із зазначенням типів даних та обмежень продемонстровано за допомогою pgAdmin4. Також наведено приклади вмісту таблиць із даними, що підтверджує коректність створення та працездатність бази даних.

Таким чином, поставлене завдання виконано повністю: база даних спроєктована, реалізована та нормалізована відповідно до вимог.

**Контрольні запитання**

**1. Сформулювати призначення діаграм типу «сутність-зв’язок».**

Вони використовуються для моделювання структури бази даних. ER-діаграма показує сутності (об’єкти предметної області), їхні атрибути та зв’язки між ними. Завдяки цьому легше спроєктувати логічну структуру бази даних ще до її реалізації.

**2. Назвати основні об’єкти схеми PostgreSQL.**

Таблиці (Tables) - зберігають дані у вигляді рядків і стовпців.

Індекси (Indexes) - прискорюють пошук та доступ до даних.

Представлення (Views) - віртуальні таблиці, що формуються запитами.

Схеми (Schemas) - логічні об’єднання об’єктів БД.

Функції та процедури (Functions, Procedures) - збережені програмні об’єкти.

Тригери (Triggers) - автоматичне виконання дій при певних подіях.

Типи даних (Data Types) - вбудовані та користувацькі.

**3. Навести приклади різних типів зв’язків у базах даних (1:1, 1:N, N:M).**

1:1 (один до одного) - кожному запису в одній таблиці відповідає рівно один запис в іншій (наприклад: людина до паспорту).

1:N (один до багатьох) - одному запису відповідає кілька записів в іншій таблиці (наприклад: факультет до студентів).

N:M (багато до багатьох) - один запис може бути пов'язаний із багатьма в іншій таблиці, і навпаки; реалізується через проміжну таблицю (наприклад: студенти до дисципліни).