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-12

Гусельніков А.О.

Перевірив:

Павловский В. І.

Київ – 2023

**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 галузі**

Обрана мною предметна галузь – Платформа для бронювання та організації екскурсій та туристичних подорожей. Вона передбачає зручний інструментарій для бронювання та організації будь-яких заходів(у даному випадку це екскурсії та туристичні подорожі).

**Опис сутностей**

Згідно цієї області для побудови бази даних було виділено наступні сутності:

1. Організатори (Оrganizers).

Атрибути: ідентифікатор організатора, ім’я організатора, прізвище організатора, електронна пошта організатора.

Призначення: збереження даних щодо організаторів заходу.

1. Туристи (Tourists).

Атрибути: ідентифікатор туриста, ім’я туриста, прізвище туриста, електронна пошта туриста.

Призначення: збереження даних щодо туристів.

1. Заходи (Events).

Атрибути: ідентифікатор заходу, назва заходу, тип заходу, дата коли відбудеться, ідентифікатор адреси.

Призначення: збереження даних щодо заходів.

1. Номери телефонів (Phone\_nums).

Атрибути: ідентифікатор номеру телефону, номер телефону, ідентифікатор туриста, якому він належить.

Призначення: збереження даних щодо контактних номерів телефонів туристів.

**Опис зв’язків між сутностями**

1. Зв’язок «Оrganizers» - «Events» є зв’язком N:M. Організатор може мати багато створених заходів, а може й не мати, у одного заходу може бути один або більше організаторів.
2. Зв’язок «Tourists» - «Events» є зв’язком N:M. Турист може забронювати місця у багатьох заходах, а може й не забронювати, захід може мати багато заброньованих місць, а може й не мати.
3. Зв’язок «Tourists» - «Phone\_nums» є зв’язком 1:N. Кожен турист має ввести хоча б один контактний номер телефону та номер телефону має належати тільки одній людині.

Графічне подання концептуальної моделі «Сутність-зв’язок» зображено на рисунку 1.

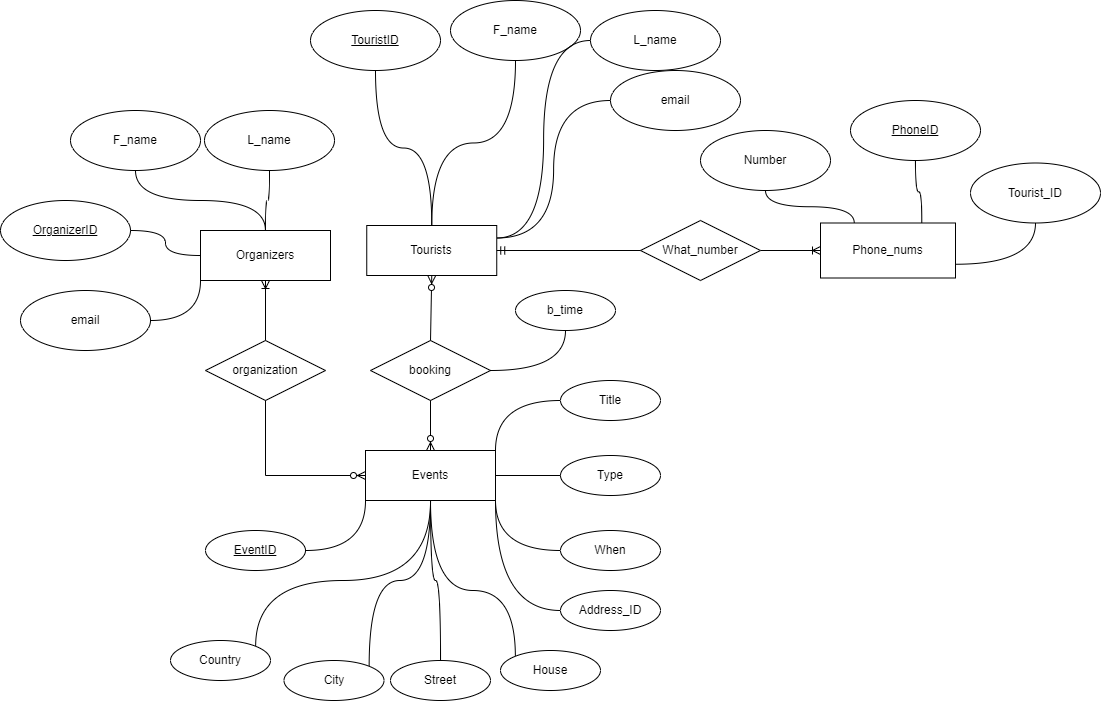


Рисунок 1 – ER-діаграма, побудована за нотацією Чена

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

Сутність Оrganizers перетворено в таблицю Оrganizers з первинним ключем ОrganizerID та атрибутами F\_name, L\_name, email.

Сутність Tourists перетворено в таблицю Tourists з первинним ключем TouristID та атрибутами F\_name, L\_name, email.

Сутність Events перетворено в таблицю Events з первинним ключем EventID та атрибутами Title, Type, Date та зовнішнім ключем Address\_ID. Та через те, що може відбуватись декілька заходів розташованих за одной адресою, створено додаткову таблицю Address з первинним ключем AddressID та атрибутами Country, City, Street, House.

Сутність Phone\_nums перетворено в таблицю Phone\_nums з первинним ключем PhoneID та атрибутом Number, та зовнішнім ключем Tourist\_ID.

В логічній моделі неможливий безпосередній зв'язок М:N, а в концептуальній моделі він існує між сутностями Оrganizers і Events, та Tourists і Events. Для представлення зв'язку між сутностями Оrganizers і Events було введено допоміжну таблицю Оrganizers/Events з первинними та зовнішніми ключами Оrganizer і Event\_o. Для представлення зв'язку між сутностями Tourists і Events було введено допоміжну таблицю Tourists/Events з первинними та зовнішніми ключами Tourist і Event\_t, та первинним ключом booking\_time.

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

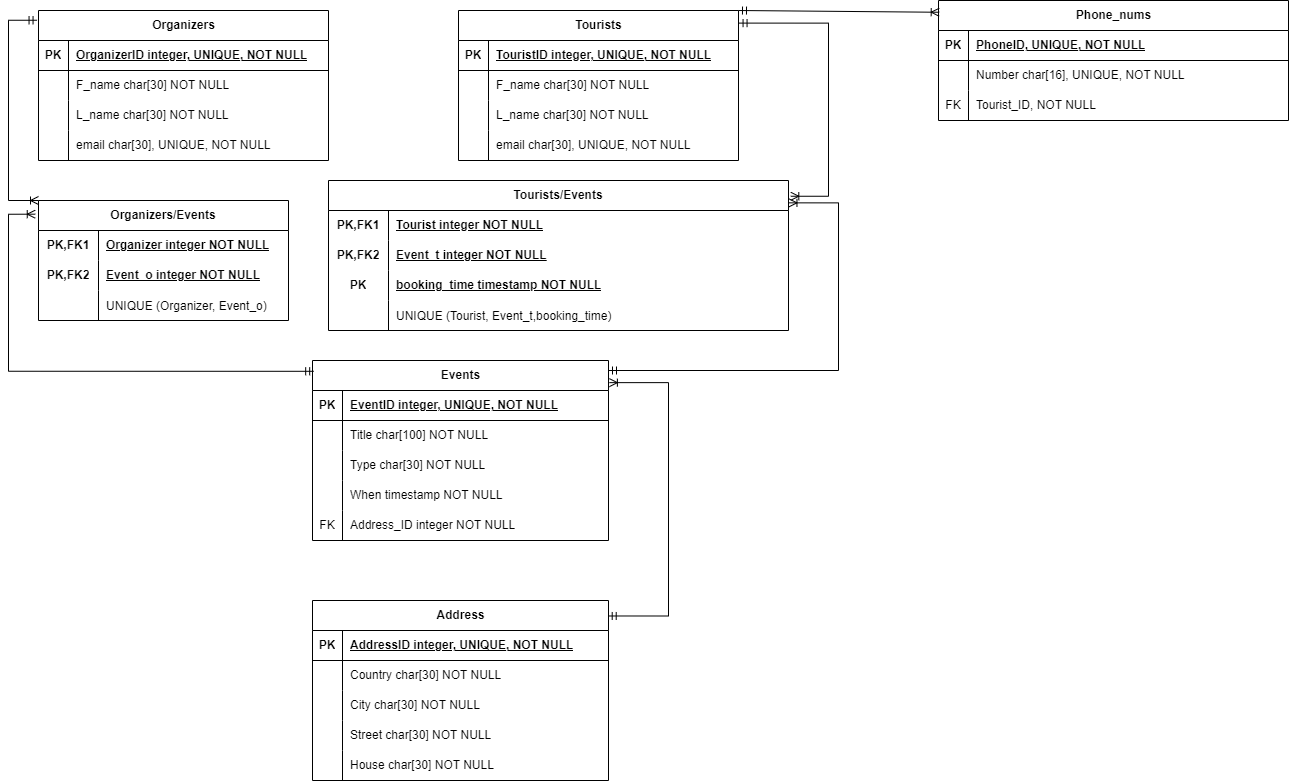


Рисунок 2 – Схема бази даних

Таблиця 1 ілюструє детальний перехід від однієї моделі до іншої.

Таблиця 1 – Опис об’єктів бази даних

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| **Оrganizers** – містить дані про організаторів. | **ОrganizerID** – унікальний ідентифікатор організатора | **integer** (число) |
| **F\_name –** ім'я організатора | **character varying** (рядок) |
| **L\_name** – прізвище організатора | **character varying** (рядок) |
| **email** – електронна пошта організатора | **character varying** (рядок) |
| **Tourists** – містить дані про туристів. | **TouristID** – унікальний ідентифікатор туриста | **integer** (число) |
| **F\_name –** ім'я туриста | **character varying** (рядок) |
| **L\_name** – прізвище туриста | **character varying** (рядок) |
| **email** – електронна пошта туриста | **character varying** (рядок) |
| **Events** – містить дані про заходи. | **EventID** – унікальний ідентифікатор заходу | **integer** (число) |
| **Title** – заголовок заходу | **character varying** (рядок) |
| **Type** – тип заходу | **character varying** (рядок) |
| **Date** – дата початку заходу | **timestamp** (дата та час) |
| **Address\_ID** – унікальний ідентифікатор адреси заходу | **integer** (число) |
| **Address** – містить дані про адресу заходу. | **AddressID** – унікальний ідентифікатор заходу | **integer** (число) |
| **Country** – країна, де буде відбуватись захід | **character varying** (рядок) |
| **City** – місто, де буде відбуватись захід | **character varying** (рядок) |
| **Street** – вулиця, де буде відбуватись захід | **character varying** (рядок) |
| **House** – будинок, де буде відбуватись захід | **character varying** (рядок) |
| **Оrganizers/Events** – містить інформацію про відповідність організаторів і створених заходів. | **Оrganizer –** унікальний ідентифікатор організатора | **integer** (число) |
| **Event\_o –** унікальний ідентифікатор заходу створеного організатором | **integer** (число) |
| **Tourists/Events** – містить інформацію про відповідність туристів і заброньованих заходів. | **Tourist** – унікальний ідентифікатор туриста | **integer** (число) |
| **Event\_t** – унікальний ідентифікатор заходу заброньованого організатором | **integer** (число) |
| **booking\_time** – дата та час бронювання | **timestamp** (дата та час) |
| **Phone\_nums –** містить інформацію про контактні номери телефонів туристів | **PhoneID –** унікальний ідентифікатор номеру телефону | **integer** (число) |
| **Number –** номер телефону | **character varying** (рядок) |
| **Tourist\_ID –** унікальний ідентифікатор туриста, який користується цим номером телефону | **integer** (число) |

**Функціональні залежності для кожної таблиці**

1. **Оrganizers:**

ОrganizerID -> F\_name, L\_name, email.

ОrganizerID -> F\_name.

ОrganizerID -> L\_name

ОrganizerID -> email.

email -> F\_name, L\_name.

email -> F\_name.

email -> L\_name.

**Атрибут email є унікальним – це ключ кандидат.**

1. **Tourists:**

TouristID -> F\_name, L\_name, email.

TouristID -> F\_name.

TouristID -> L\_name

TouristID -> email.

email -> F\_name, L\_name.

email -> F\_name.

email -> L\_name.

**Атрибут email є унікальним – це ключ кандидат.**

1. **Events:**

EventID -> Title, Type, Date, Address\_ID.

EventID -> Title.

EventID -> Type.

EventID -> Date.

EventID -> Address\_ID.

1. **Address:**

AddressID -> Country, City, Street, House.

AddressID -> Country.

AddressID -> City.

AddressID -> Street.

AddressID -> House.

1. **Оrganizers/Events:**

Оrganizer, Event\_o -> Event\_o.

Оrganizer, Event\_o -> Оrganizer.

1. **Tourists/Events:**

Tourist, Event\_t, booking\_time - > Tourist.

Tourist, Event\_t, booking\_time - > Event\_t.

Tourist, Event\_t, booking\_time - > booking\_time.

1. **Phone\_nums:**

PhoneID -> Number, Tourist\_ID.

PhoneID -> Number.

PhoneID -> Tourist\_ID.

Number -> Tourist\_ID.

**Атрибут Number є унікальним – це ключ кандидат.**

**Відповідність схеми нормальним формам**

Схема відповідає 1НФ, тому що:

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

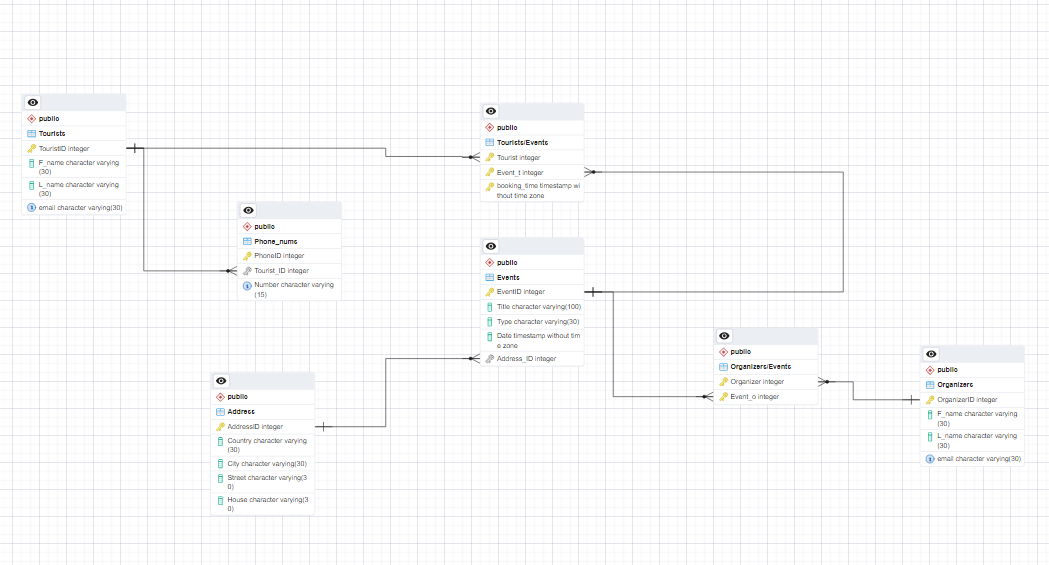
Схема відповідає 2НФ, тому що:

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

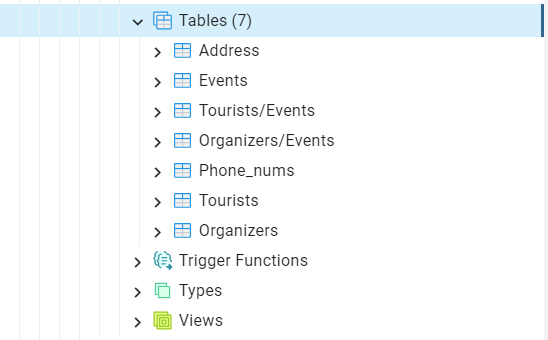
Схема відповідає 3НФ, тому що:

* Схема перебуває в 2НФ;
* Кожен неключовий атрибут функціонально залежить від усього ключа, відсутня транзитивна залежність.

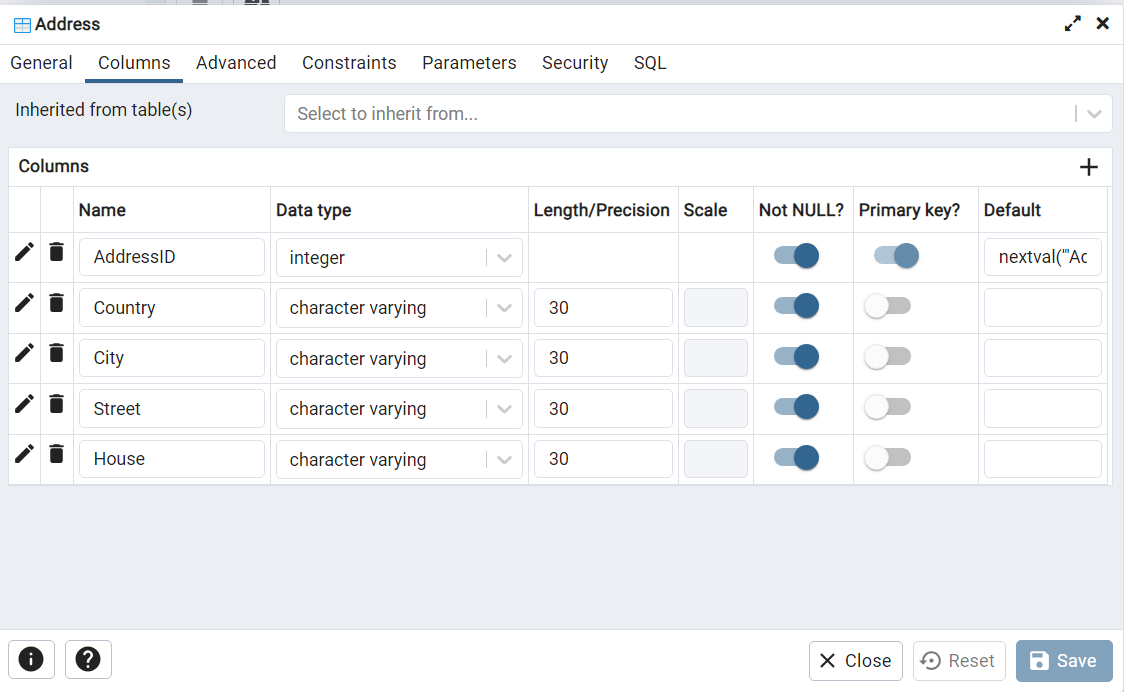
Рисунок 3 - Схема бази даних у pgAdmin 4

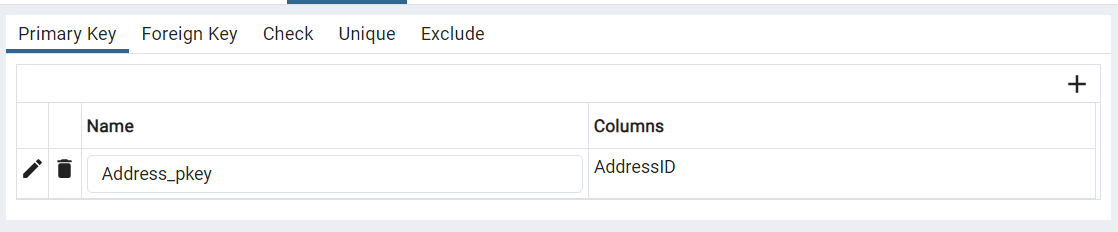
****

**Таблиці бази даних у pgAdmin4**

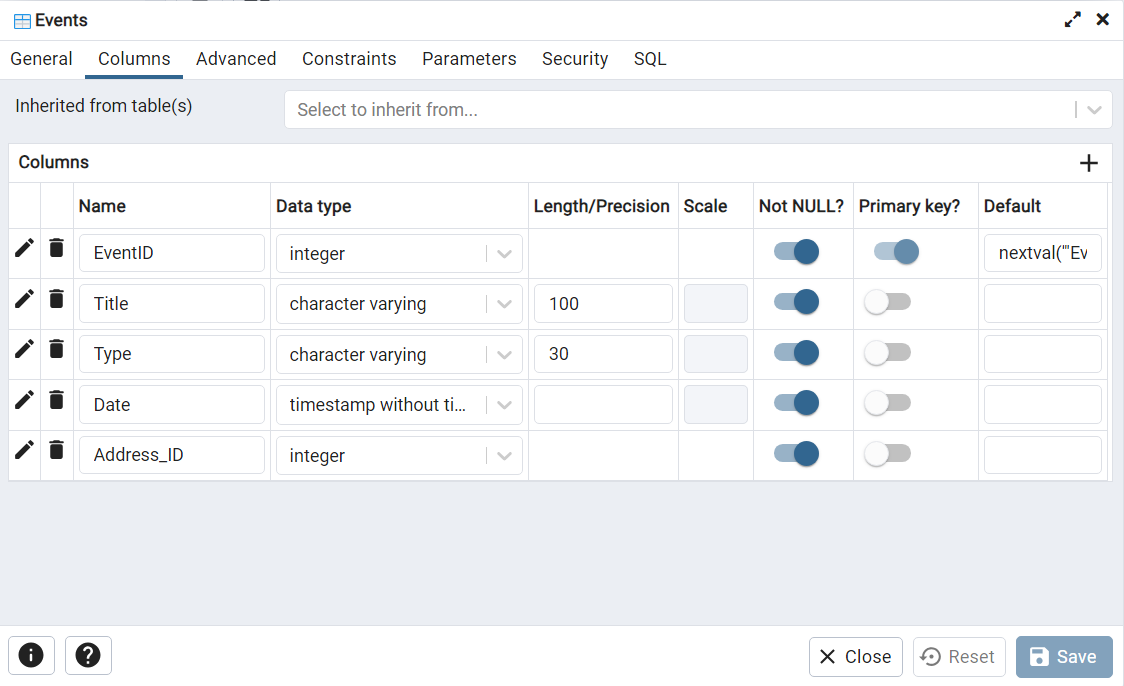
****

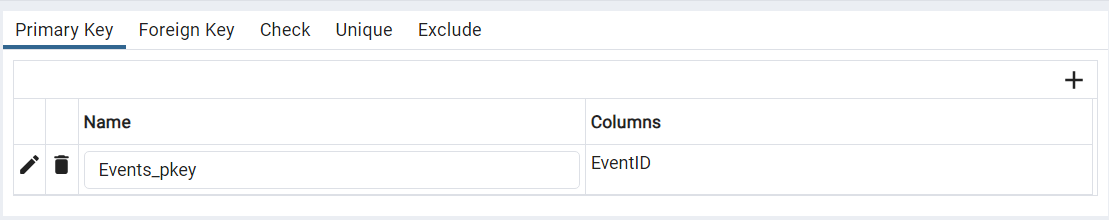
**Address**

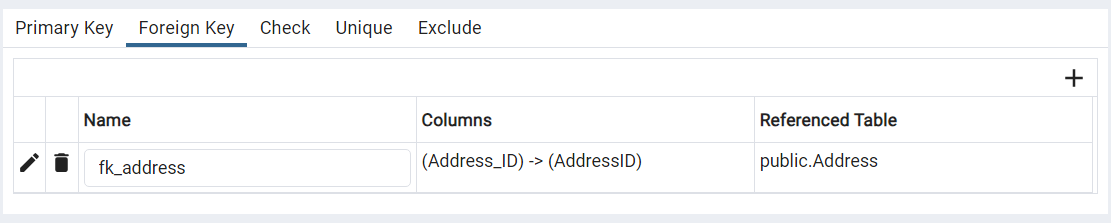
****

****

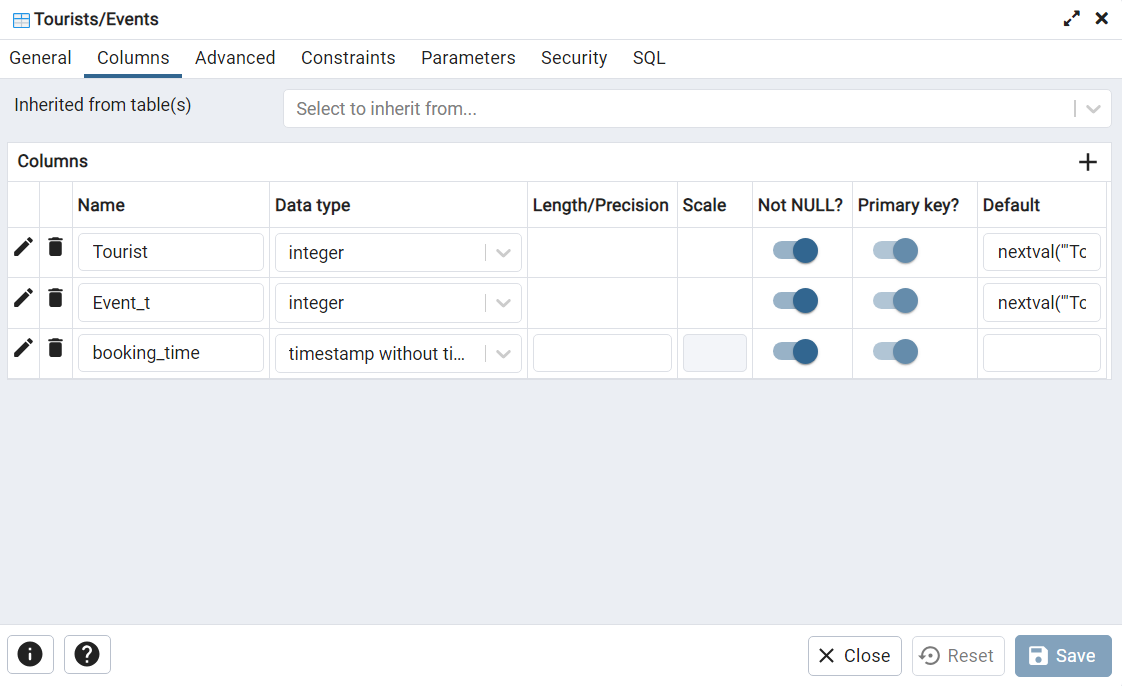
**Events**

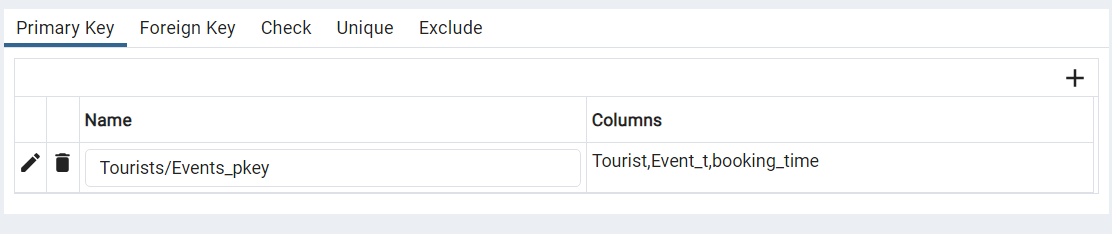
****

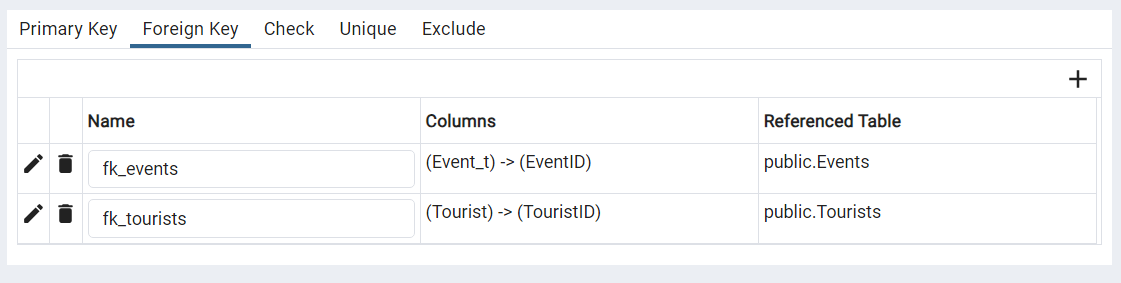
****

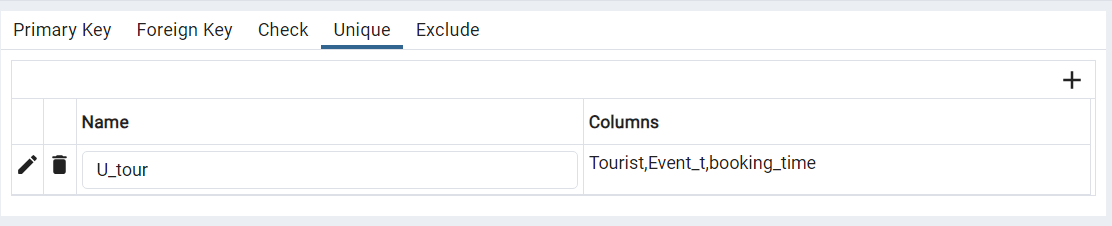
****

**Tourists/Events**

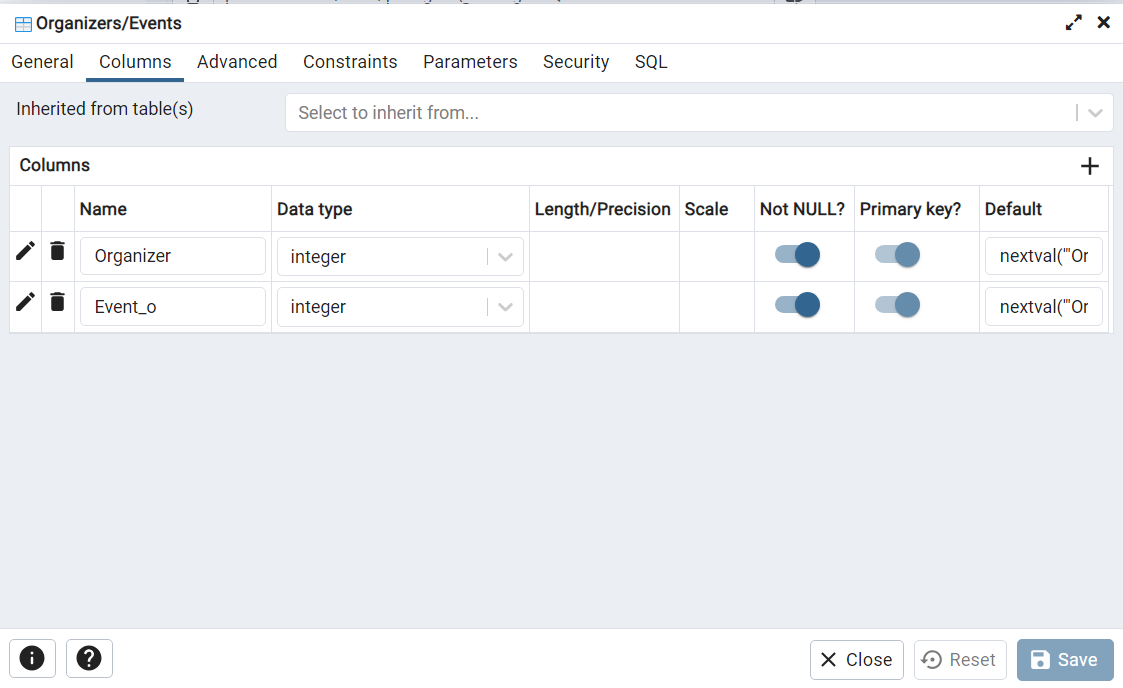
****

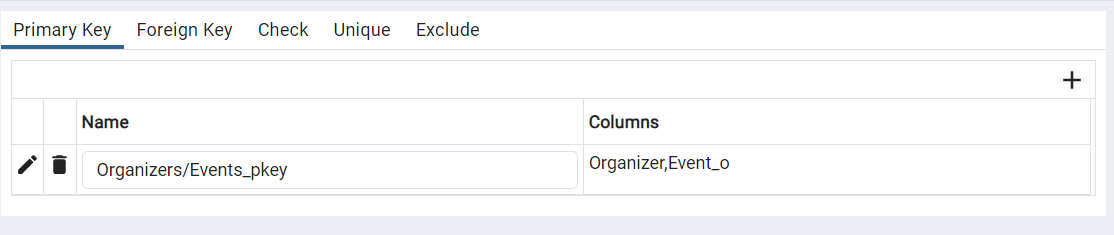
****

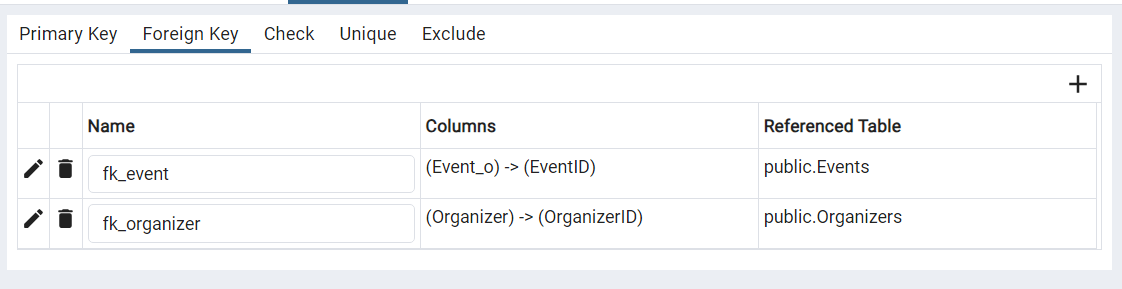
****

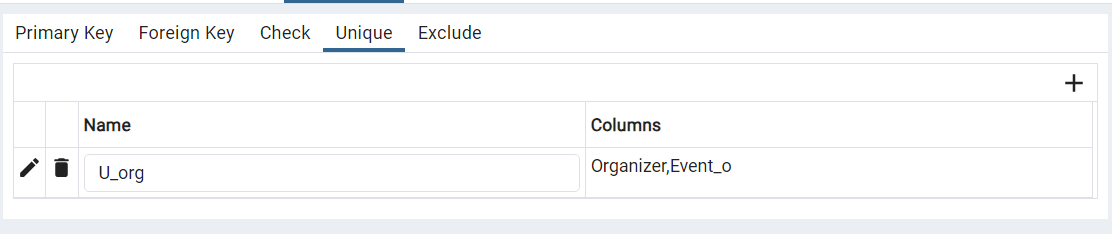
****

**Оrganizers/Events**

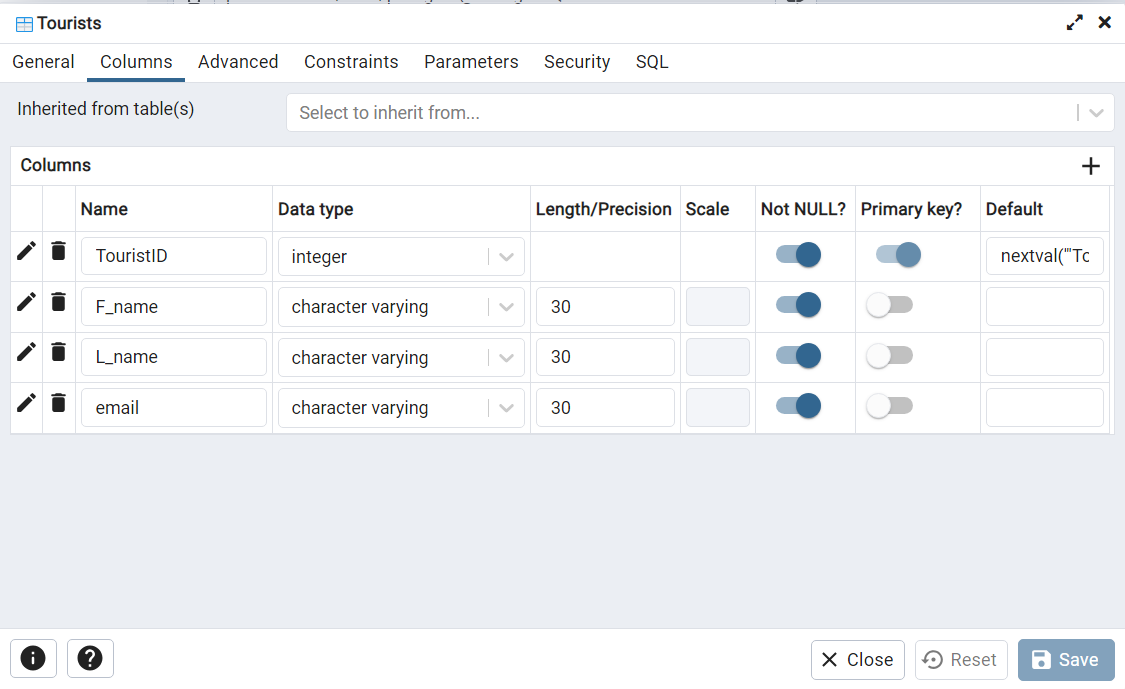
****

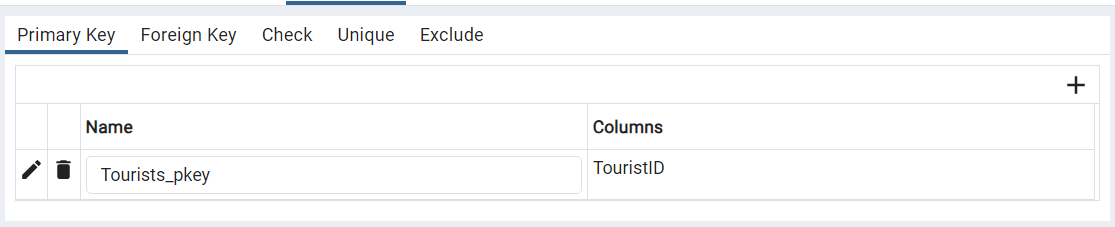
****

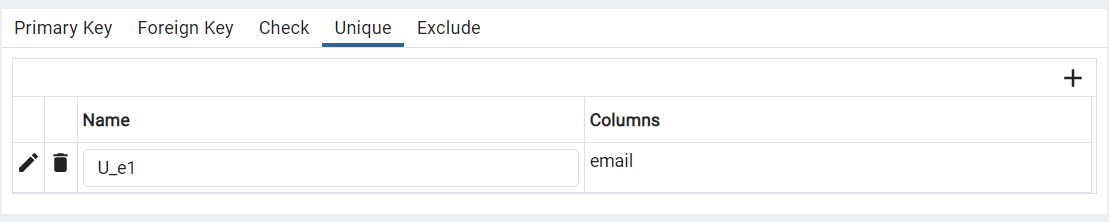
****

****

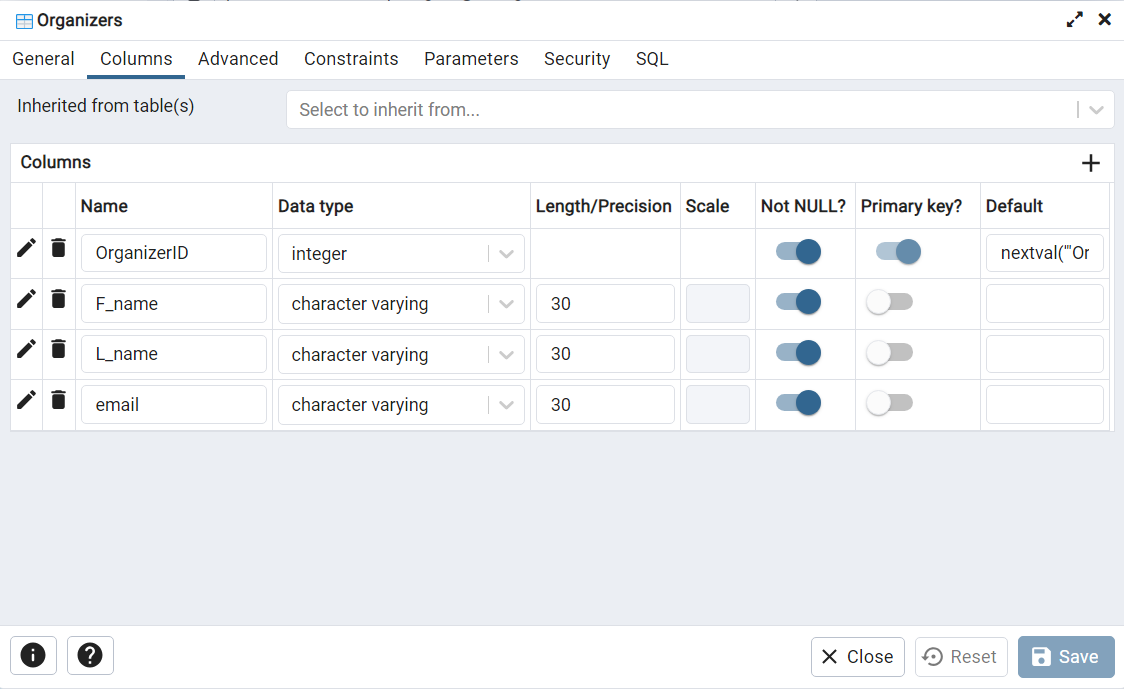
**Tourists**

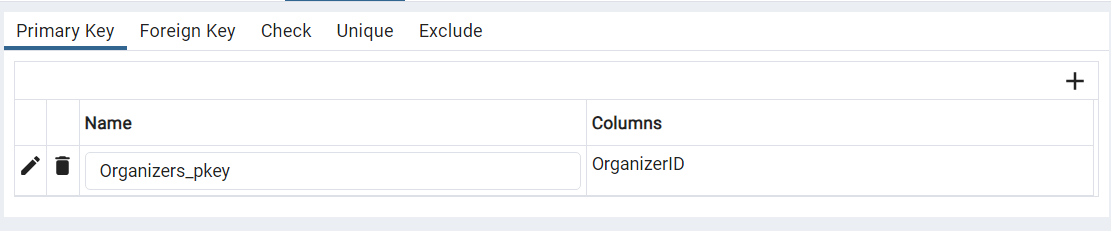
****

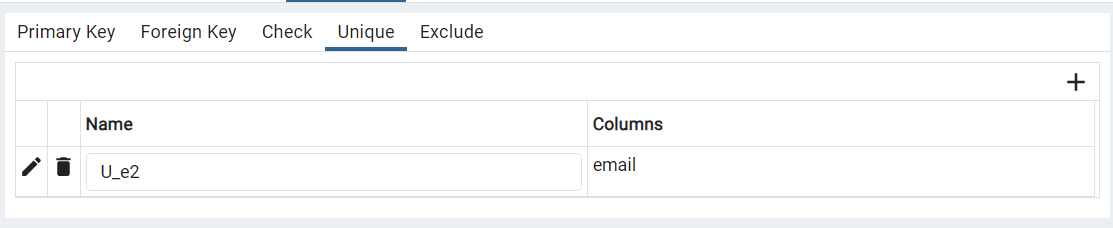
****

****

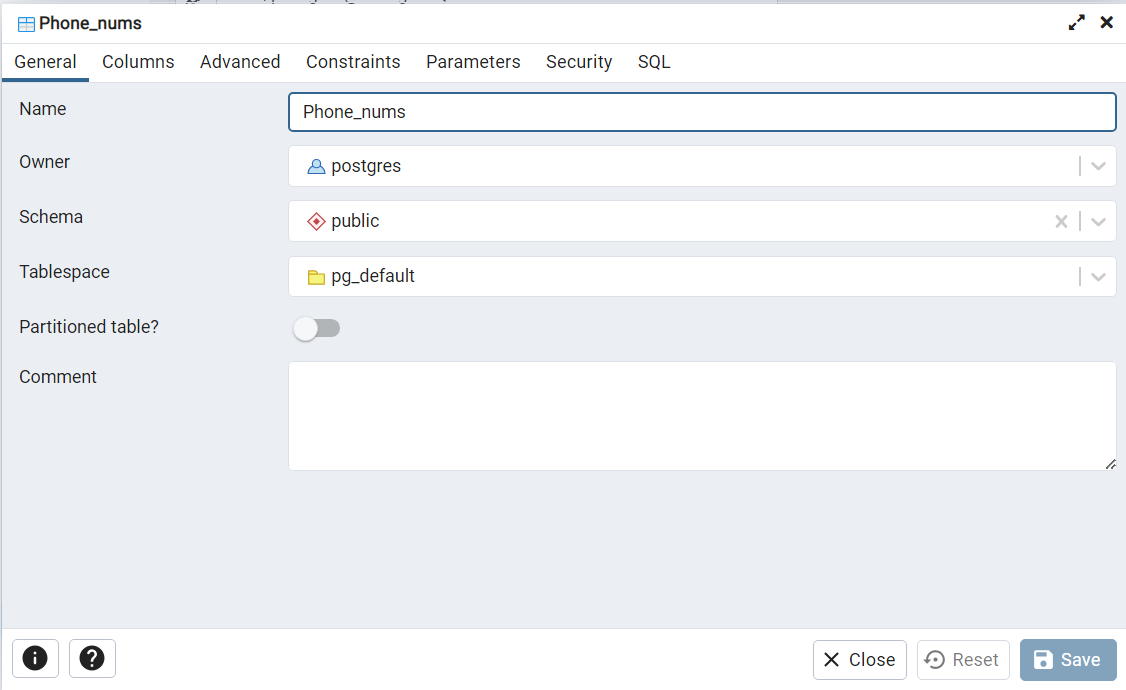
**Оrganizers**

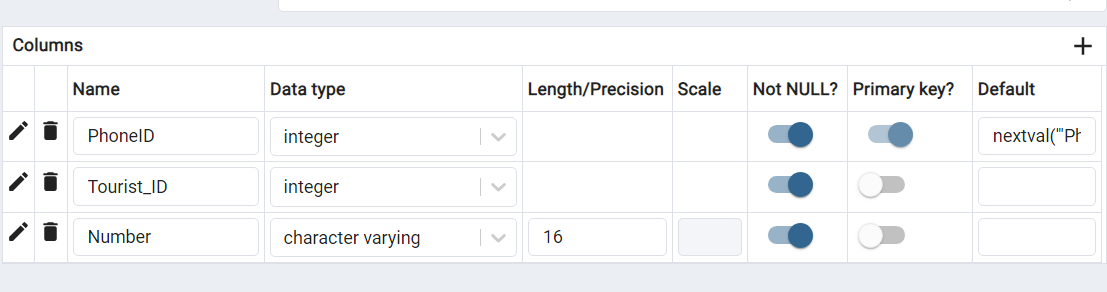
****

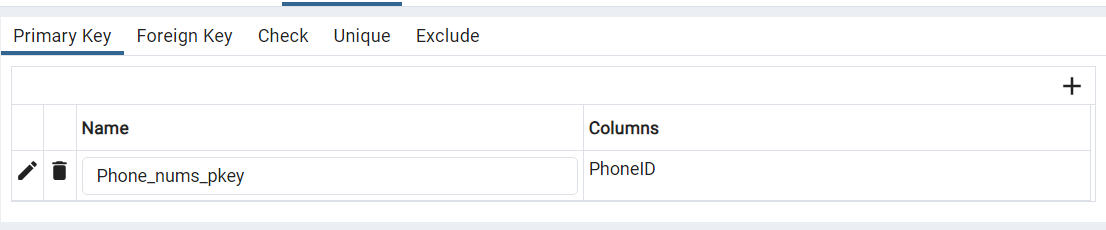
****

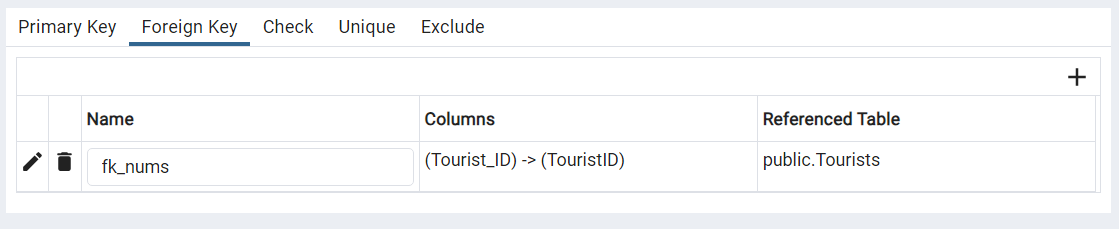
****

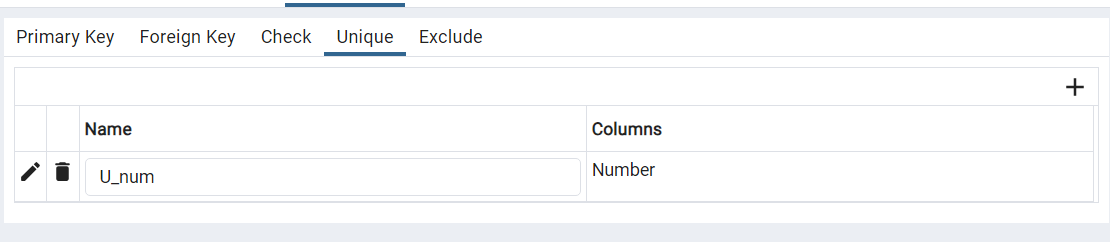
**Phone\_nums**

****

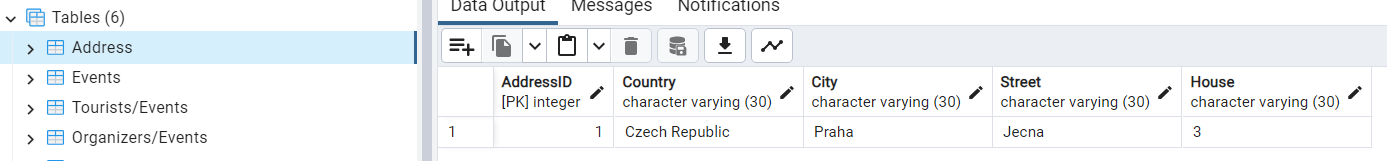
****

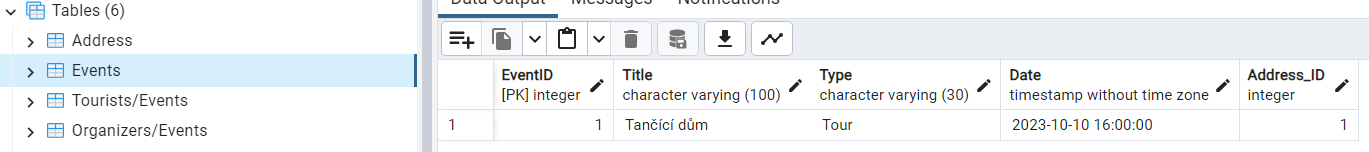
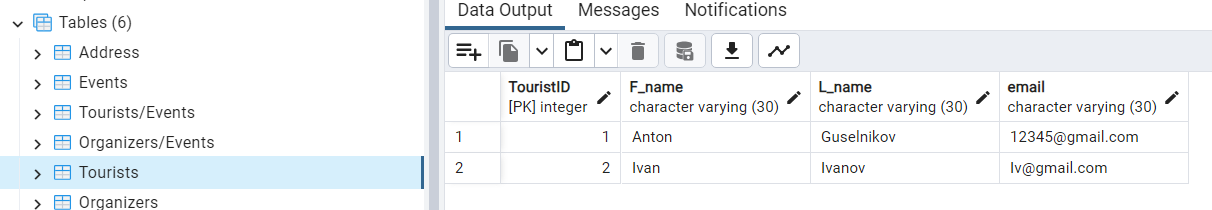
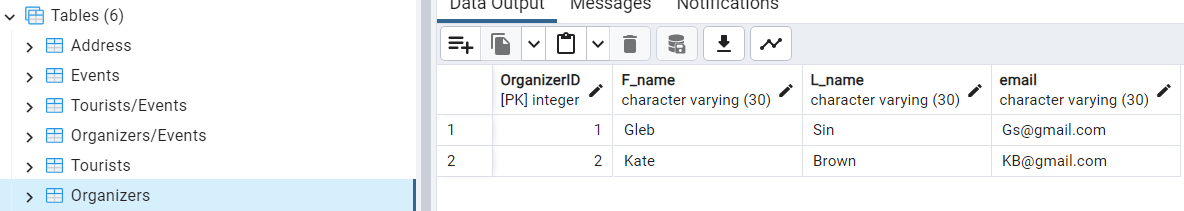
****

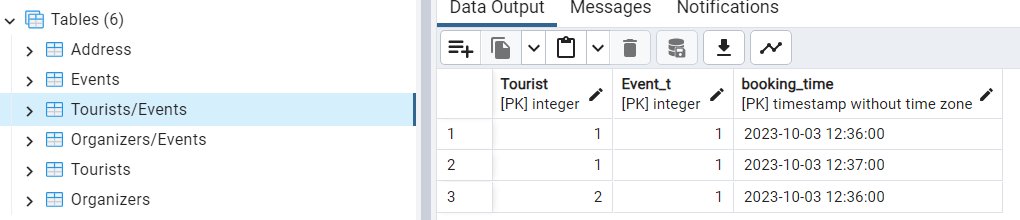
****

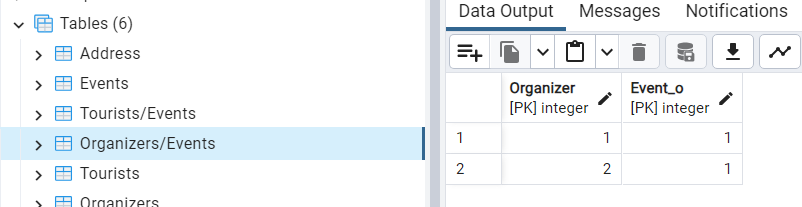
****

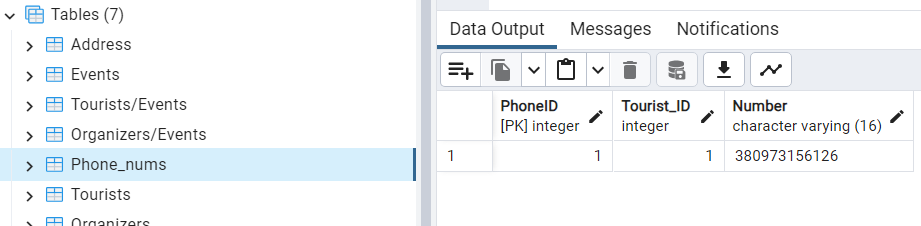
**Таблиці заповнені даними (уривки знімків екрану pgAdmin4)**

****

**** ****

****

****

****

**Таблиці в коді SQL**

-- Table: public.Address

-- DROP TABLE IF EXISTS public."Address";

CREATE TABLE IF NOT EXISTS public."Address"

(

"AddressID" integer NOT NULL DEFAULT nextval('"Address\_AddressID\_seq"'::regclass),

"Country" character varying(30) COLLATE pg\_catalog."default" NOT NULL,

"City" character varying(30) COLLATE pg\_catalog."default" NOT NULL,

"Street" character varying(30) COLLATE pg\_catalog."default" NOT NULL,

"House" character varying(30) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Address\_pkey" PRIMARY KEY ("AddressID")

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Address"

OWNER to postgres;

-- Table: public.Events

-- DROP TABLE IF EXISTS public."Events";

CREATE TABLE IF NOT EXISTS public."Events"

(

"EventID" integer NOT NULL DEFAULT nextval('"Events\_EventID\_seq"'::regclass),

"Title" character varying(100) COLLATE pg\_catalog."default" NOT NULL,

"Type" character varying(30) COLLATE pg\_catalog."default" NOT NULL,

"Date" timestamp without time zone NOT NULL,

"Address\_ID" integer NOT NULL,

CONSTRAINT "Events\_pkey" PRIMARY KEY ("EventID"),

CONSTRAINT fk\_address FOREIGN KEY ("Address\_ID")

REFERENCES public."Address" ("AddressID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Events"

OWNER to postgres;

-- Table: public.Tourists/Events

-- DROP TABLE IF EXISTS public."Tourists/Events";

CREATE TABLE IF NOT EXISTS public."Tourists/Events"

(

"Tourist" integer NOT NULL DEFAULT nextval('"Tourists/Events\_Tourist\_seq"'::regclass),

"Event\_t" integer NOT NULL DEFAULT nextval('"Tourists/Events\_Event\_t\_seq"'::regclass),

booking\_time timestamp without time zone NOT NULL,

CONSTRAINT "Tourists/Events\_pkey" PRIMARY KEY ("Tourist", "Event\_t", booking\_time),

CONSTRAINT "U\_tour" UNIQUE ("Tourist", "Event\_t", booking\_time),

CONSTRAINT fk\_events FOREIGN KEY ("Event\_t")

REFERENCES public."Events" ("EventID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT fk\_tourists FOREIGN KEY ("Tourist")

REFERENCES public."Tourists" ("TouristID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Tourists/Events"

OWNER to postgres;

-- Table: public.Оrganizers/Events

-- DROP TABLE IF EXISTS public."Оrganizers/Events";

CREATE TABLE IF NOT EXISTS public."Оrganizers/Events"

(

"Оrganizer" integer NOT NULL DEFAULT nextval('"Оrganizers/Events\_Оrganizer\_seq"'::regclass),

"Event\_o" integer NOT NULL DEFAULT nextval('"Оrganizers/Events\_Event\_seq"'::regclass),

CONSTRAINT "Оrganizers/Events\_pkey" PRIMARY KEY ("Оrganizer", "Event\_o"),

CONSTRAINT "U\_org" UNIQUE ("Оrganizer", "Event\_o"),

CONSTRAINT fk\_event FOREIGN KEY ("Event\_o")

REFERENCES public."Events" ("EventID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT fk\_organizer FOREIGN KEY ("Оrganizer")

REFERENCES public."Оrganizers" ("ОrganizerID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Оrganizers/Events"

OWNER to postgres;

-- Table: public.Phone\_nums

-- DROP TABLE IF EXISTS public."Phone\_nums";

CREATE TABLE IF NOT EXISTS public."Phone\_nums"

(

"PhoneID" integer NOT NULL DEFAULT nextval('"Phone\_nums\_PhoneID\_seq"'::regclass),

"Tourist\_ID" integer NOT NULL,

"Number" character varying(16) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Phone\_nums\_pkey" PRIMARY KEY ("PhoneID"),

CONSTRAINT "U\_num" UNIQUE ("Number"),

CONSTRAINT fk\_nums FOREIGN KEY ("Tourist\_ID")

REFERENCES public."Tourists" ("TouristID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Phone\_nums"

OWNER to postgres;

-- Table: public.Tourists

-- DROP TABLE IF EXISTS public."Tourists";

CREATE TABLE IF NOT EXISTS public."Tourists"

(

"TouristID" integer NOT NULL DEFAULT nextval('"Tourists\_TouristID\_seq"'::regclass),

"F\_name" character varying(30) COLLATE pg\_catalog."default" NOT NULL,

"L\_name" character varying(30) COLLATE pg\_catalog."default" NOT NULL,

email character varying(30) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Tourists\_pkey" PRIMARY KEY ("TouristID"),

CONSTRAINT "U\_e1" UNIQUE (email)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Tourists"

OWNER to postgres;

-- Table: public.Оrganizers

-- DROP TABLE IF EXISTS public."Оrganizers";

CREATE TABLE IF NOT EXISTS public."Оrganizers"

(

"ОrganizerID" integer NOT NULL DEFAULT nextval('"Оrganizers\_ОrganizerID\_seq"'::regclass),

"F\_name" character varying(30) COLLATE pg\_catalog."default" NOT NULL,

"L\_name" character varying(30) COLLATE pg\_catalog."default" NOT NULL,

email character varying(30) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Оrganizers\_pkey" PRIMARY KEY ("ОrganizerID"),

CONSTRAINT "U\_e2" UNIQUE (email)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Оrganizers"

OWNER to postgres;