Зображення, що містить текст

Автоматично згенерований опис

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

# Кафедра системного програмування і спеціалізованих комп’ютерних систем

**Лабораторна робота №1**

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

з дисципліни

**«Бази даних і засоби управління»**

Виконав: студент ІII курсу

ФПМ групи КВ-93

Піскун Андрій

Перевірив:

Павловський В. І.

Київ – 2021

*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.

**Перелік сутностей з описом їх призначення**

Кінотеатр (Cinema) з атрибутами: назва, код, адреса кінотеатру. Призначена для збереження даних про кінотеатри, в яких купуються квитки.

Фільм (Movie) з атрибутами: назва, код, рейтинг. Призначена для збереження даних про фільми, що показують.

Сеанс (Session) з атрибутами: час, код, вартість. Призначена для збереження даних про сеанси фільмів.

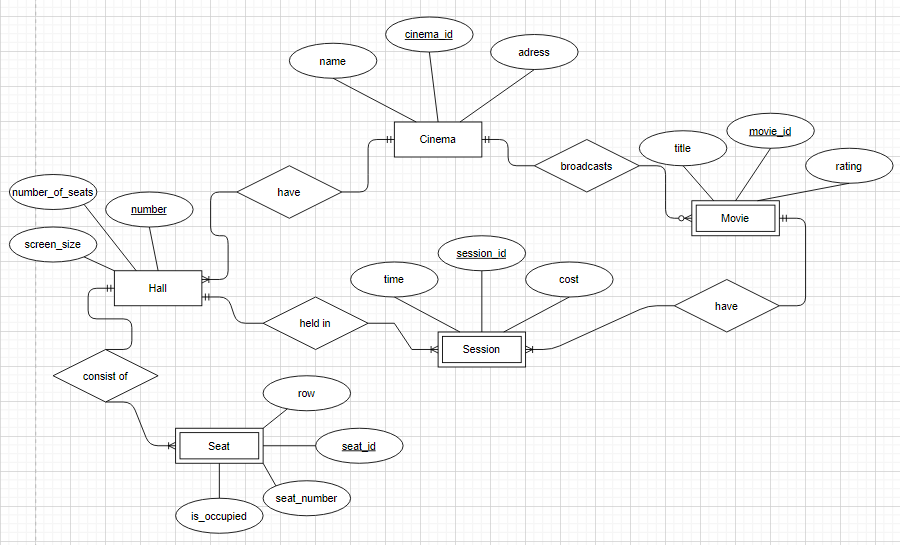
Зал (Hall) з атрибутами: код, розмір екрану, кількість місць. Призначена для збереження даних про зали в кінотеатрі.

Місце (Seat) з атрибутами: ряд, код, місце, зайнятість місця. Призначена для збереження даних про місце в залі.

**Опис зв’язків**

Один кінотеатр може транслювати багато фільмів, фільм може мати декілька сеансів, кінотеатр може мати багато залів, в залі може проводитись декілька сеансів, як і зал має багато місць, тому зв’язки 1:N.

**Модель “сутність-зв’язок” предметної галузі кінотеатр**

 Рисунок 1. ER-діаграма, побудована за нотацією Чена

Побудовано за допомогою веб додатку draw.io

**Перетворення моделі у схему баз даних**

Сутність ‘Cinema’ перетворено у таблицю ‘Cinema’.

Сутність ‘Movie’ перетворено у таблицю ‘Movie’, а зв’язок 1:N із сутністю ‘Cinema’ зумовив появу зовнішнього ключа cinema\_id.

Сутність ‘Hall’ перетворено у таблицю ‘Hall’, а зв’язок 1:N із сутністю ‘Cinema’ зумовив появу зовнішнього ключа cinema\_id.

Сутність ‘Seat’ перетворено у таблицю ‘Seat’, а зв’язок 1:N із сутністю ‘Hall’ зумовив появу зовнішнього ключа number.

Сутність ‘Session’ перетворено у таблицю ‘Session’, зв’язок 1:N із сутністю ‘Movie’ зумовив появу зовнішнього ключа movie\_id, зв’язок 1:N із сутністю ‘Hall’ зумовив появу зовнішнього ключа number.

Зображення, що містить текст, у приміщенні

Автоматично згенерований опис

Рисунок 2. Схема бази даних у таблицях

Побудовано за допомогою веб додатку draw.io

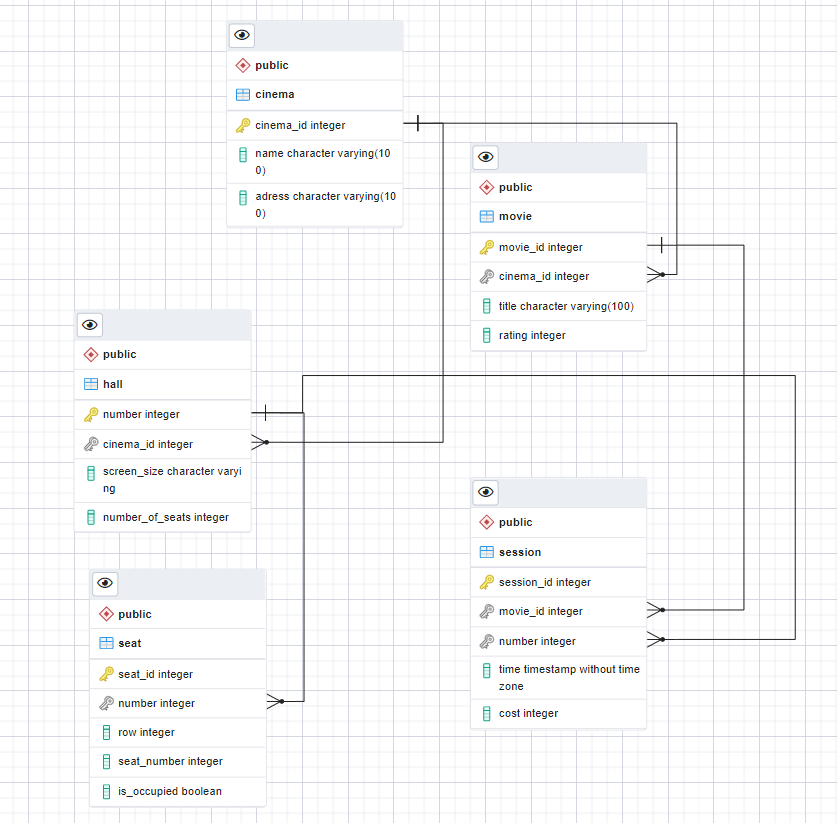


Рисунок 3. Схема бази даних у pgAdmin 4

**Опис структури БД**

Cinema – містить дані про кінотеатр

* cinema\_id (числовий) – унікальний ідентифікатор кінотеатру
* name (текстовий) – назва кінотеатру
* address (текстовий) – адреса кінотеатру

Movie – містить дані про фільм

* movie\_id (числовий) – унікальний ідентифікатор фільму
* cinema\_id (числовий) – ідентифікатор кінотеатру, в якому транслюється фільм
* title (текстовий) – назва фільму
* rating (числовий) – рейтинг фільму

Hall – містить дані про зал

* number (числовий) – унікальний ідентифікатор залу, а також його номер
* cinema\_id (числовий) – ідентифікатор кінотеатру, в якому знаходиться зал
* screen\_size (текстовий) – розмір екрану у залі
* number\_of\_seats (числовий) – кількість місць у залі

Seat – містить дані про місце в залі

* seat\_id (числовий) – унікальний ідентифікатор місця
* number (числовий) – ідентифікатор залу, в якому знаходиться місце
* row (числовий) – номер ряду
* seat\_number (числовий) – номер місця в ряду
* is\_occupied (двійковий) – інформація про те, чи зайняте місце

Session – містить дані про сеанс фільму

* session\_id (числовий) – унікальний ідентифікатор сеансу
* movie\_id (числовий) – ідентифікатор фільму, сеанс якого проводиться
* number (числовий) – ідентифікатор залу, в якому проводиться сеанс
* time (мітка часу) – час, о котрій проводиться сеанс
* cost (числовий) – вартість сеансу

**Функціональні залежності**

Cinema:

cinema\_id -> name, address

cinema\_id -> name (назва пов’язана з ідентифікатором)

cinema\_id -> address (адреса пов’язана з ідентифікатором)

name -> address

address -> name

cinema\_id -> name -> address (включає ключ)

cinema\_id -> address -> name (включає ключ)

Movie:

movie\_id -> title, rating

movie\_id -> title

movie\_id -> rating

title -> rating

movie\_id -> title -> rating (включає ключ)

Hall:

number -> screen\_size, number\_of\_seats

number -> screen\_size

number -> number\_of\_seats

Seat:

seat\_id -> row, seat\_number, is\_occupied

seat\_id -> row

seat\_id -> seat\_number

seat\_id -> is\_occupied

seat\_number -> is\_occupied

seat\_id -> seat\_number -> is\_occupied (includes PK)

Session:

session\_id -> time, cost

session\_id -> time

session\_id -> cost

Схема відповідає 1НФ, тому що:

* Атрибути мають унікальні значення
* В кожному стовпці зберігаються дані одного типу.
* В кожній комірці зберігається атомарне (скалярне) значення.

Схема відповідає 2НФ, тому що:

* Відповідає 1НФ
* Кожен не ключовий атрибут функціонально залежний від ключового

Схема відповідає 3НФ, тому що:

* Відповідає 3НФ
* Залежності в одній таблиці тільки від основного ключа

**SQL-текст опису БД**

BEGIN;

CREATE TABLE IF NOT EXISTS public.cinema

(

cinema\_id integer NOT NULL,

name character varying(100) NOT NULL,

adress character varying(100) NOT NULL,

PRIMARY KEY (cinema\_id)

);

CREATE TABLE IF NOT EXISTS public.hall

(

"number" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

cinema\_id integer NOT NULL,

screen\_size character varying NOT NULL,

number\_of\_seats integer NOT NULL,

PRIMARY KEY ("number")

);

CREATE TABLE IF NOT EXISTS public.movie

(

movie\_id integer NOT NULL GENERATED BY DEFAULT AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

cinema\_id integer NOT NULL,

title character varying(100) NOT NULL,

rating integer NOT NULL,

PRIMARY KEY (movie\_id)

);

CREATE TABLE IF NOT EXISTS public.seat

(

seat\_id integer NOT NULL,

"number" integer NOT NULL,

"row" integer NOT NULL,

seat\_number integer NOT NULL,

is\_occupied boolean NOT NULL,

PRIMARY KEY (seat\_id)

);

CREATE TABLE IF NOT EXISTS public.session

(

session\_id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

movie\_id integer NOT NULL,

"number" integer NOT NULL,

"time" timestamp without time zone NOT NULL,

cost integer NOT NULL,

PRIMARY KEY (session\_id)

);

ALTER TABLE public.hall

ADD FOREIGN KEY (cinema\_id)

REFERENCES public.cinema (cinema\_id)

NOT VALID;

ALTER TABLE public.movie

ADD FOREIGN KEY (cinema\_id)

REFERENCES public.cinema (cinema\_id)

NOT VALID;

ALTER TABLE public.seat

ADD FOREIGN KEY ("number")

REFERENCES public.hall ("number")

NOT VALID;

ALTER TABLE public.session

ADD FOREIGN KEY (movie\_id)

REFERENCES public.movie (movie\_id)

NOT VALID;

ALTER TABLE public.session

ADD FOREIGN KEY ("number")

REFERENCES public.hall ("number")

NOT VALID;

END;

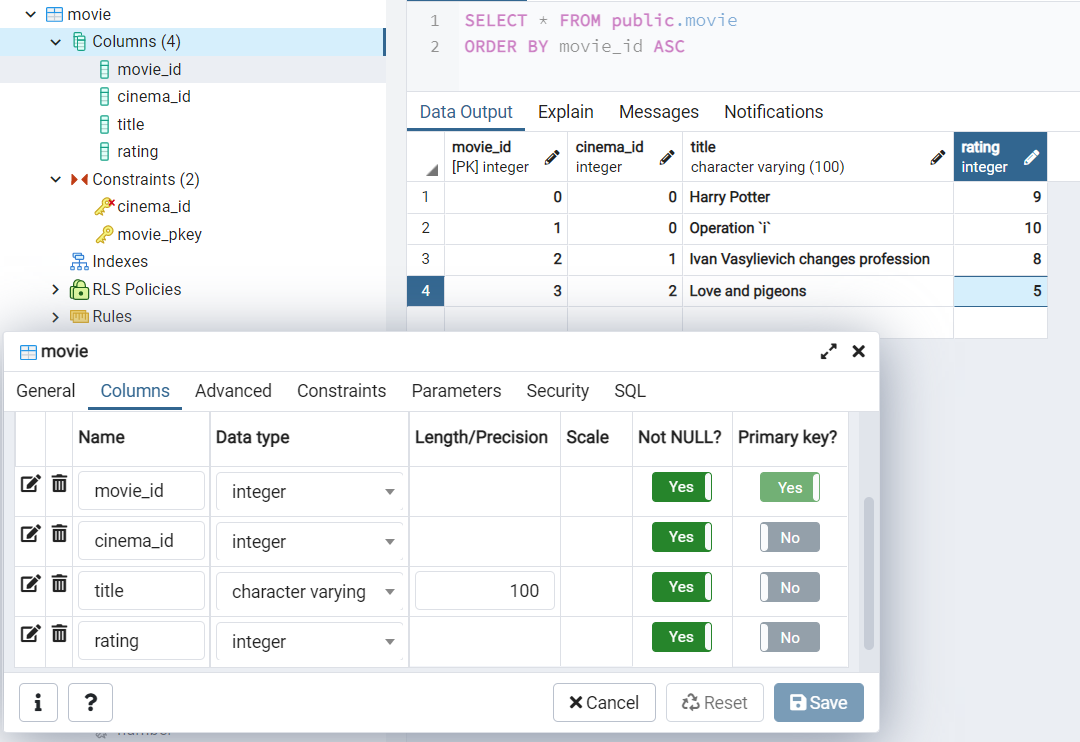
**Копії екрану з pgAdmin4**

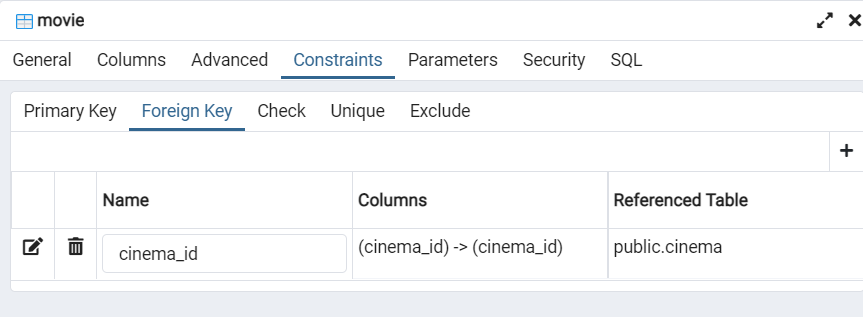
**Cinema**

Зображення, що містить стіл

Автоматично згенерований опис

**Movie**

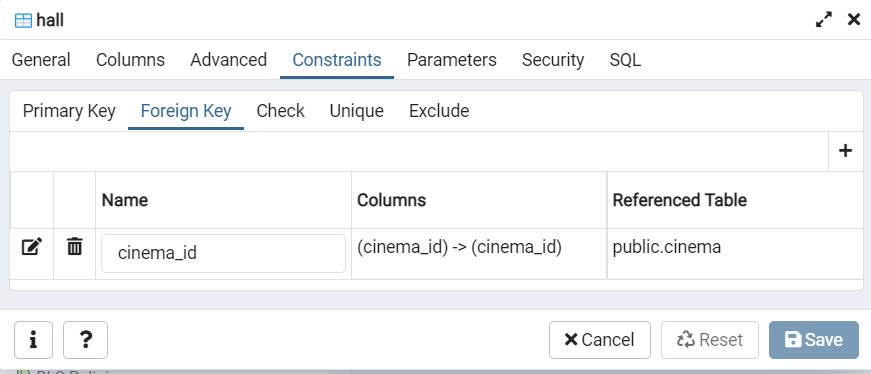




**Hall**

Зображення, що містить стіл

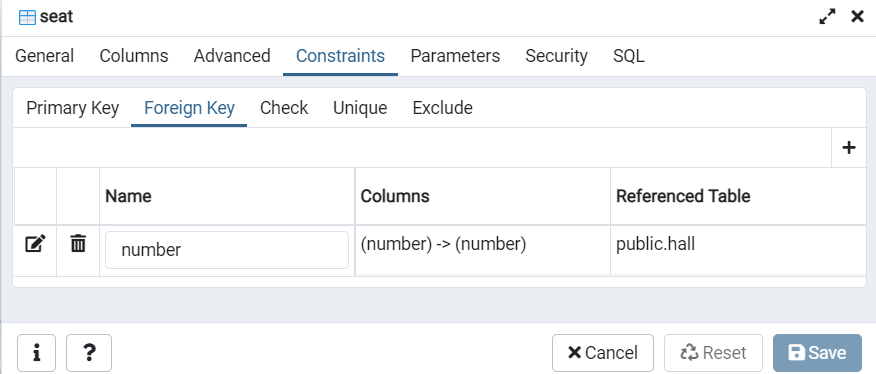
Автоматично згенерований опис



**Seat**

Зображення, що містить стіл

Автоматично згенерований опис



**Session**

Зображення, що містить стіл

Автоматично згенерований опис

Зображення, що містить текст

Автоматично згенерований опис