

**Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки**

**Лабораторна робота №2**

з дисципліни  
«Бази даних»

Виконала:

студентка групи ІМ-42  
Шалак Анастасія Володимирівна  
номер у списку групи: 30

Перевірив:

Русінов В. В.

Київ 2025

## Перетворення ER-діаграми на схему PostgreSQL

### Цілі

- Написати SQL DDL-інструкції для створення кожної таблиці з вашої ERD в PostgreSQL.
- Вказати відповідні типи даних для кожного стовпця, вибрати первинний ключ для кожної таблиці та визначити будь-які необхідні зовнішні ключі, обмеження UNIQUE, NOT NULL, CHECK або DEFAULT.
- Вставити зразки рядків (принаймні 3–5 рядків на таблицю) за допомогою INSERT INTO.
- Нарешті, протестувати все в pgAdmin (або іншому клієнті PostgreSQL), щоб переконатися, що таблиці та дані завантажуються правильно.

### Хід виконання роботи

- SQL-скрипт(и) у звіті з усіма вашими операторами CREATE TABLE та INSERT по порядку. Ви можете додавати коментарі до SQL-файлу для ідентифікації таблиць.

```
CREATE TABLE Hall (
```

```
    hall_id SERIAL PRIMARY KEY,
```

```
    hall_name VARCHAR(255) NOT NULL UNIQUE
```

```
);
```

```
INSERT INTO Hall (hall_name) VALUES
```

```
('Зал 1'),
```

```
('Зал 2'),
```

```
('VIP Зал');
```

```
CREATE TABLE Movie (  
    movie_id SERIAL PRIMARY KEY,  
    title VARCHAR(255) NOT NULL UNIQUE,  
    budget NUMERIC(15,2),  
    description TEXT,  
    release_date DATE,  
    box_office NUMERIC(15,2),  
    duration INT,  
    tagline VARCHAR(255),  
    average_rating REAL  
);
```

```
INSERT INTO Movie (title, budget, description, release_date, box_office, duration,  
tagline, average_rating) VALUES  
  
('Inception', 160000000, 'Sci-fi thriller by Christopher Nolan', '2010-07-16', 830000000,  
148, 'Your mind is the scene of the crime', 8.8),  
  
('Interstellar', 165000000, 'Epic space adventure about love and time', '2014-11-07',  
700000000, 169, 'Mankind was born on Earth. It was never meant to die here.', 8.6),  
  
('The Dark Knight', 185000000, 'Batman faces Joker in Gotham', '2008-07-18',  
1000000000, 152, 'Why so serious?', 9.0);
```

```
CREATE TABLE Genres (  
    genre_id SERIAL PRIMARY KEY,  
    genre_name VARCHAR(255) NOT NULL UNIQUE  
);
```

```
INSERT INTO Genres (genre_name) VALUES
```

```
('Action'),
```

```
('Science Fiction'),
```

```
('Drama');
```

```
CREATE TABLE Movies_Genres (
```

```
    movie_id INT NOT NULL,
```

```
    genre_id INT NOT NULL,
```

```
    PRIMARY KEY (movie_id, genre_id),
```

```
        FOREIGN KEY (movie_id) REFERENCES Movie(movie_id) ON DELETE  
CASCADE,
```

```
        FOREIGN KEY (genre_id) REFERENCES Genres(genre_id) ON DELETE  
CASCADE
```

```
);
```

```
INSERT INTO Movies_Genres (movie_id, genre_id) VALUES
```

```
(1, 2),
```

```
(2, 2),
```

```
(3, 1);
```

```
CREATE TABLE Showings (
```

```
    showing_id SERIAL PRIMARY KEY,
```

```
    show_time TIMESTAMP NOT NULL,
```

```
price NUMERIC(8,2) NOT NULL  
);
```

```
INSERT INTO Showings (show_time, price) VALUES  
( '2025-10-21 18:00', 150.00),  
( '2025-10-21 20:00', 200.00),  
( '2025-10-22 16:00', 120.00);
```

```
CREATE TABLE Schedule (  
    schedule_id SERIAL PRIMARY KEY,  
    hall_id INT NOT NULL,  
    showing_id INT NOT NULL,  
    movie_id INT NOT NULL,  
    FOREIGN KEY (hall_id) REFERENCES Hall(hall_id) ON DELETE CASCADE,  
    FOREIGN KEY (showing_id) REFERENCES Showings(showing_id) ON DELETE  
    CASCADE,  
    FOREIGN KEY (movie_id) REFERENCES Movie(movie_id) ON DELETE  
    CASCADE  
);
```

```
INSERT INTO Schedule (hall_id, showing_id, movie_id) VALUES  
(1, 1, 1),  
(2, 2, 2),  
(3, 3, 3);
```

```
CREATE TABLE Clients (  
    client_id SERIAL PRIMARY KEY,  
    last_name VARCHAR(100) NOT NULL,  
    first_name VARCHAR(100) NOT NULL  
);
```

```
INSERT INTO Clients (last_name, first_name) VALUES  
  
('Іваненко', 'Олег'),  
  
('Петренко', 'Марія'),  
  
('Коваль', 'Андрій');
```

```
CREATE TABLE Tickets (  
    ticket_id SERIAL PRIMARY KEY,  
    show_date TIMESTAMP NOT NULL,  
    schedule_id INT NOT NULL,  
    client_id INT NOT NULL,  
    seat VARCHAR(10) NOT NULL,  
  
    FOREIGN KEY (schedule_id) REFERENCES Schedule(schedule_id) ON DELETE  
CASCADE,  
  
    FOREIGN KEY (client_id) REFERENCES Clients(client_id) ON DELETE  
CASCADE  
);
```

```
INSERT INTO Tickets (show_date, schedule_id, client_id, seat) VALUES
('2025-10-21 18:00', 1, 1, 'A5'),
('2025-10-21 20:00', 2, 2, 'B3'),
('2025-10-22 16:00', 3, 3, 'C7');
```

– Короткий письмовий звіт, що підсумовує вашу остаточну схему. Перелічіть кожену таблицю з її стовпцями та ключами, а також поясніть будь-які важливі обмеження або припущення (наприклад, «У таблиці Orders customer\_id – це зовнішній ключ, що посилається на Customers(customer\_id)). За потреби додайте діаграму вашої схеми.

### *Hall*

Поле	Тип	Ключі/обмеження	Опис
hall_id	SERIAL	PRIMARY KEY	Унікальний ідентифікатор залу
hall_name	VARCHAR(255)	NOT NULL, UNIQUE	Назва залу, кожен зал має унікальну назву

### *Movie*

Поле	Тип	Ключі/обмеження	Опис
movie_id	SERIAL	PRIMARY KEY	Ідентифікатор фільму
title	VARCHAR(255)	NOT NULL, UNIQUE	Назва фільму, кожна назва має бути унікальною
budget	NUMERIC(15,2)		Бюджет фільму
description	TEXT		Опис
release_date	DATE		Дата виходу
box_office	NUMERIC(15,2)		Касові збори

duration	INT		Тривалість у хвилинах
tagline	VARCHAR(255)		Рекламний слоган
average_rating	REAL		Середній рейтинг

### *Genres*

Поле	Тип	Ключі/обмеження	Опис
genre_id	SERIAL	PRIMARY KEY	Ідентифікатор жанру
genre_name	VARCHAR(255)	NOT NULL, UNIQUE	Назва жанру, не повторюється

### *Movies\_Genres*

Поле	Тип	Ключі/обмеження	Опис
movie_id	INT	FOREIGN KEY – Movie(movie_id)	Ідентифікатор фільму
genre_id	INT	FOREIGN KEY – Genres(genre_id)	Ідентифікатор жанру

### *Showings*

Поле	Тип	Ключі/обмеження	Опис
showing_id	SERIAL	PRIMARY KEY	Ідентифікатор сеансу
show_time	TIMESTAMP	NOT NULL	Дата та час показу
price	NUMERIC(8,2)	NOT NULL	Вартість квитка

### *Schedule*

Поле	Тип	Ключі/обмеження	Опис
schedule_id	SERIAL	PRIMARY KEY	Ідентифікатор запису розкладу
hall_id	INT	FOREIGN KEY – Hall(hall_id)	Зал показу
showing_id	INT	FOREIGN KEY – Showings(showing_id)	Сеанс (час і ціна)



movie_id	INT	FOREIGN KEY – Movie(movie_id)	Показуваний фільм
----------	-----	----------------------------------	-------------------

**Обмеження:** видалення фільму, залу або сеансу призводить до каскадного видалення розкладу.

### *Clients*

Поле	Тип	Ключі/обмеження	Опис
client_id	SERIAL	PRIMARY KEY	Ідентифікатор клієнта
last_name	VARCHAR(100)	NOT NULL	Прізвище
first_name	VARCHAR(100)	NOT NULL	Ім'я

### *Tickets*

Поле	Тип	Ключі/обмеження	Опис
ticket_id	SERIAL	PRIMARY KEY	Ідентифікатор квитка
show_date	TIMESTAMP	NOT NULL	Дата та час показу
schedule_id	INT	FOREIGN KEY – Schedule(schedule_id)	Посилання на розклад
client_id	INT	FOREIGN KEY – Clients(client_id)	Покупець квитка
seat	VARCHAR(10)	NOT NULL	Номер місця

**Обмеження:** кожен квиток належить одному клієнту і відповідає одному розкладу.

– Доказ того, що таблиці були заповнені (це може бути частина SQL-файлу або скріншоти результатів запиту). Переконайтеся, що кожна таблиця містить щонайменше 3–5 рядків.



Query

Query History

1

SELECT client\_id, last\_name, first\_name

2

FROM public.clients;

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	client_id [PK] integer	last_name character varying (100)	first_name character varying (100)
1	1	Іваненко	Олег
2	2	Петренко	Марія
3	3	Коваль	Андрій

– Ви можете додати знімки екрана pgAdmin, що показують ваші таблиці, інструмент запитів

▼

🗄️ cinema

>

👥 Casts

>

📖 Catalogs

>

🔄 Event Triggers

>

📦 Extensions

>

🗄️ Foreign Data Wrappers

>

💬 Languages

>

📡 Publications

▼

🔗 Schemas (1)

▼

📊 Tables (8)

>

📊 clients

>

📊 genres

>

📊 hall

>

📊 movie

>

📊 movies\_genres

>

📊 schedule

>

📊 showings

>

📊 tickets

>

🔄 Trigger Functions

>

📦 Types

