

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**Кафедра системного програмування та 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**»

Виконав студент групи : КВ-23

ПІБ : Буц Аліса Сергіївна

Перевірив:

**Київ 2024**

**Варіант 47**

***“Система обліку виконавців та виступів на фестивалях”***

*У звіті щодо пункту №1 завдання має бути:*

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

*У звіті щодо пункту №2 завдання має бути:*

* 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.**

*У звіті щодо пункту №3 завдання має бути:*

* 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.

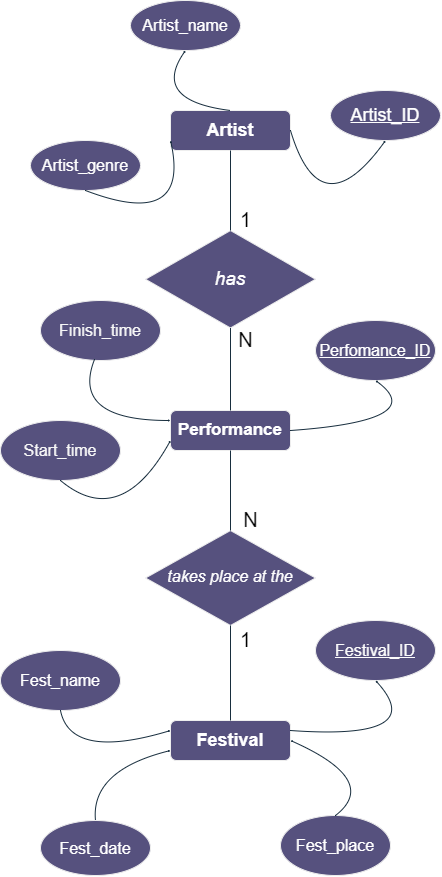
*У звіті щодо пункту №4 завдання має бути:*

* 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**!

*1 Пункт*

Модель “Система обліку виконавців та виступів на фестивалях” має такі сутності та зв’язки :

1. Сутність “Festival” (Фестиваль) з атрибутами “Festival\_ID” (primary key) “Fest\_name”, “Fest\_date”, “Fest\_place”.
2. Сутність “Artist” (Виконавець) з атрибутами “Artist\_ID” (primary key), “Artist\_name”, “Artist\_genre”
3. Сутність “Performance” (Виступ) з атрибутами “Performance\_ID” (primary key), “Festival\_ID” (foreign key), “Artist\_ID”(foreign key), “Start\_time”, “Finish\_time”



Назва нотації : нотація Чена.

**Зв'язок між Festival і Performance:**

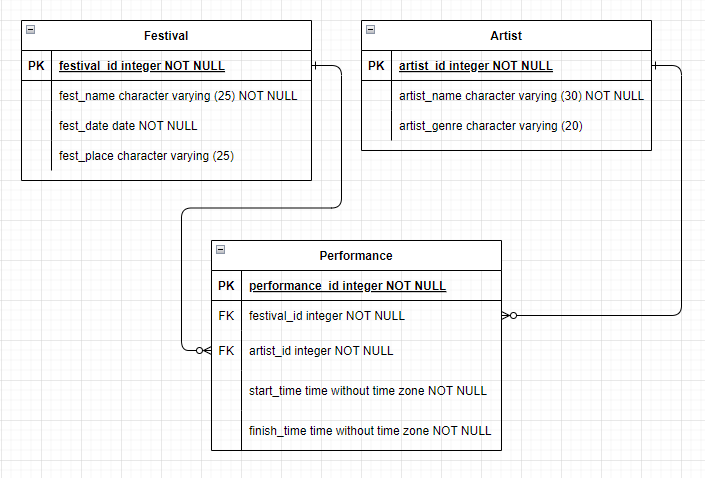
* Зв'язок є **обов'язковим** для сутності **Performance**. Кожен виступ пов'язаний з певним фестивалем.
* Зв'язок є **необов'язковим** для сутності **Festival**. Фестиваль може існувати без виступів, але коли створюється виступ, він обов'язково повинен бути прив'язаний до фестивалю.

**Зв'язок між Artist і Performance:**

* Зв'язок є **обов'язковим** для сутності **Performance**. Кожен виступ повинен мати одного виконавця.
* Зв'язок є **необов'язковим** для сутності **Artist**. Виконавець може існувати в базі даних без участі у виступах, але кожен виступ обов'язково має бути прив'язаний до виконавця.

*2 Пункт*

Перетворена ER-діаграма у таблиці БД.

**

*3 Пункт*

**Artist-Performance :**

Кожен виконавець може брати участь у багатьох виступах, але кожен виступ проводить 1 виконавець.

**Festival-Performance :**

Кожен фестиваль може складатись з багатьох виступів, але кожен виступ належить тільки 1 фестивалю.

**Непрямий зв'язок Artist-Festival :**

* Artist (artist) виконує виступ (Performance).
* Festival включає виступи.
* Таблиця Performance містить зв'язок між Artist\_ID і Festival\_ID, тобто кожен виступ належить певному фестивалю і має виконавця.

**Функціональні залежності :**

1. Festival

Festival\_ID → Fest\_name, Fest\_date, Fest\_place

Знаючи Festival\_ID, знаємо назву фестивалю, дату і місце проведення.

1. Artist

Artist\_ID → Artist\_name, Artist\_genre

Знаючи Artist\_ID, знаємо імя(або назву) виконавця та жанр.

1. Performance

Performance\_ID → Festival\_ID, Artist\_ID, Start\_time, Finish\_time

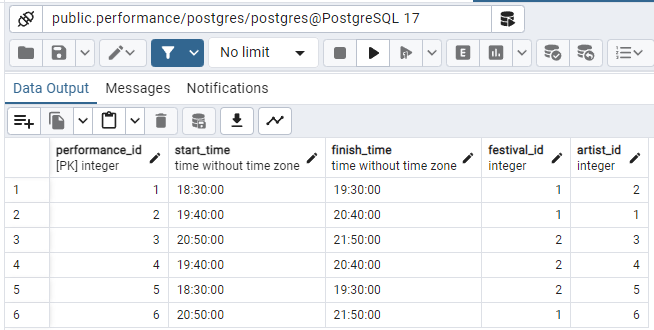
Знаючи Performance\_ID, знаємо фестиваль, де виступ проходить (Festival\_ID), виконавця (Artist\_ID), час початку і завершення.

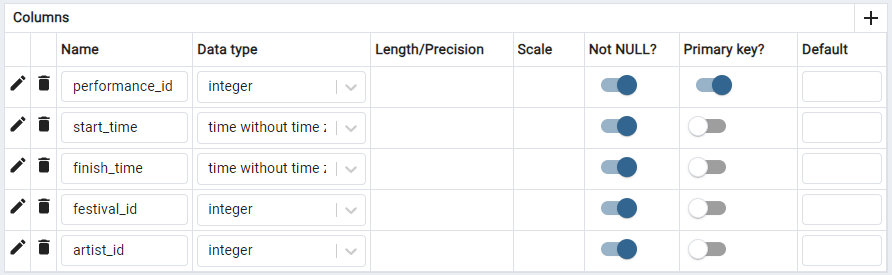
Схема бази даних відповідає 1НФ, тому що передбачає лише одне значення атрибута у кожній комірці.

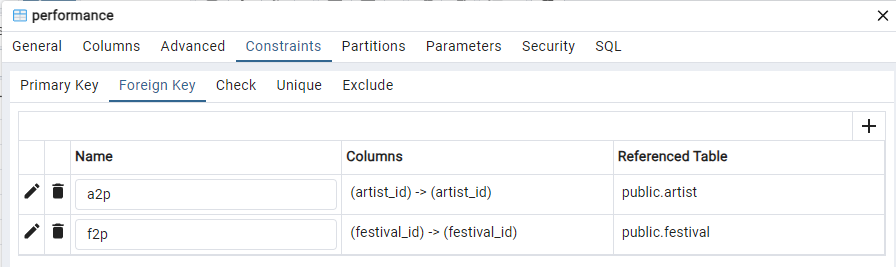
Схема відповідає 2НФ, тому що немає ключів, які складаються з двох і більше атрибутів.

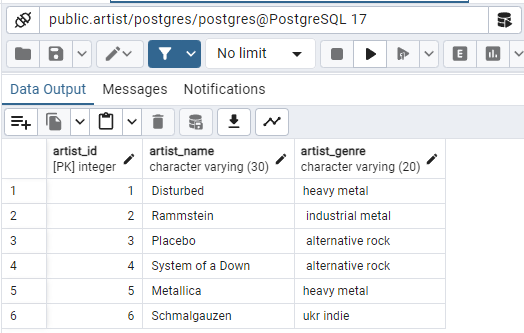
Схема відповідає 3НФ, тому що немає транзитивних функціональних залежностей між ключовими та неключовими атрибутами.

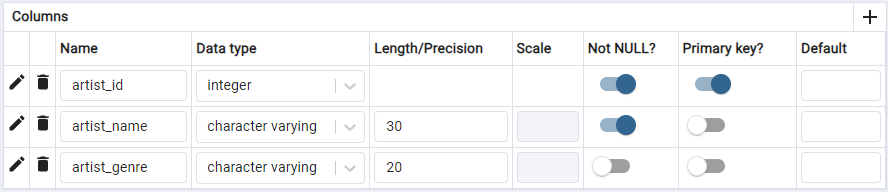
*4 Пункт*

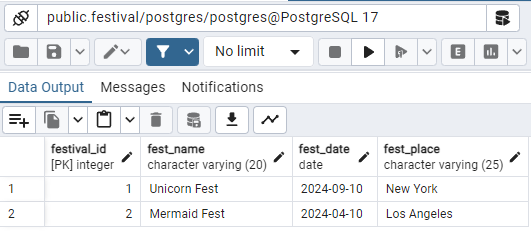
**

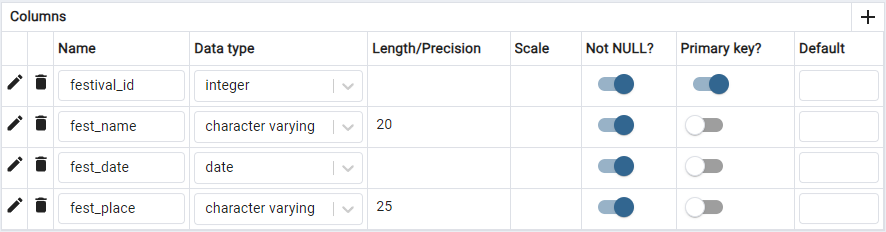
**

**

**

**

**

**

*Посилання на репозиторій :* [*https://github.com/ButsAlice/BD.git*](https://github.com/ButsAlice/BD.git)