**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 ”**

**Група: КВ-03**

**Виконав: Колесник В.В**

**Київ – 2022**

**Обрана предметна галузь**

Сервіс для прослуховування музики.

**Пункт №1**

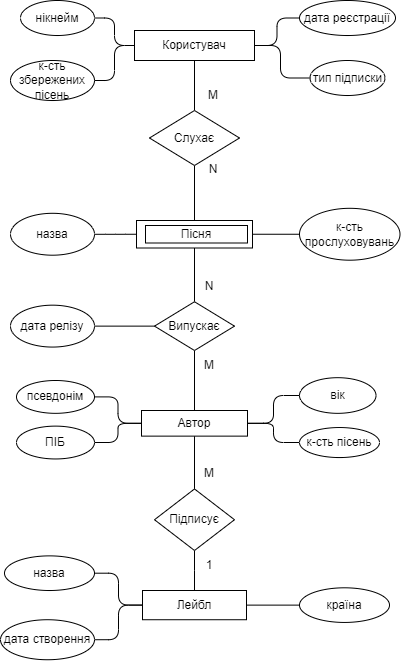
Сутність “користувач” – користувач сервісу. Може слухати пісні. Атрибути: нікнейм, дата реєстрації, к-сть збережених пісень, тип підписки (free, student, family, premium).

Сутність “пісня” – пісні, які користувач може слухати. Обов’язково написані автором(авторами). Атрибути: назва, к-сть прослуховувань.

Сутність “автор” – автор пісень (у однієї пісні може бути декілька авторів). Випускає їх. Атрибути: ПІБ, псевдонім (сценічне ім’я), вік, к-сть пісень.

Сутність “лейбл” – бренд, що може підписувати авторів на контракт. Прийнято, що один виконавець може бути лише на одному лейблі (хоча інколи бувають і декілька, але це рідкість. Тому приймемо за умову, що лише один, щоб показати у роботі зв’язок 1:M). Атрибути: назва, дата створення, країна.

**Графічне зображення моделі (нотація Чена)**



**Пункт №2**

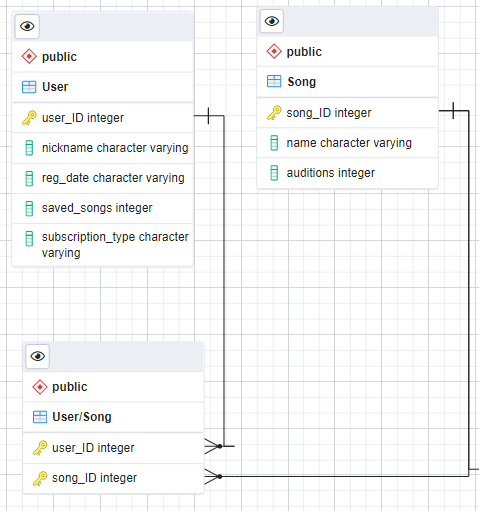
Сутність “користувач” була перетворена в таблицю User.

Сутність “пісня” була перетворена в таблицю Song.

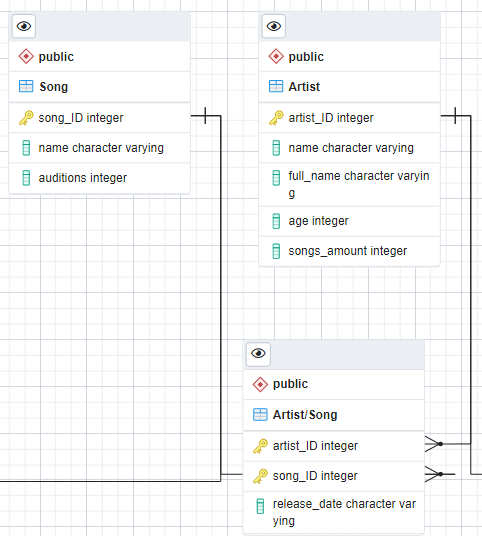
Сутність “автор” була перетворена в таблицю Artist.

Сутність “лейбл” була перетворена в таблицю Label.

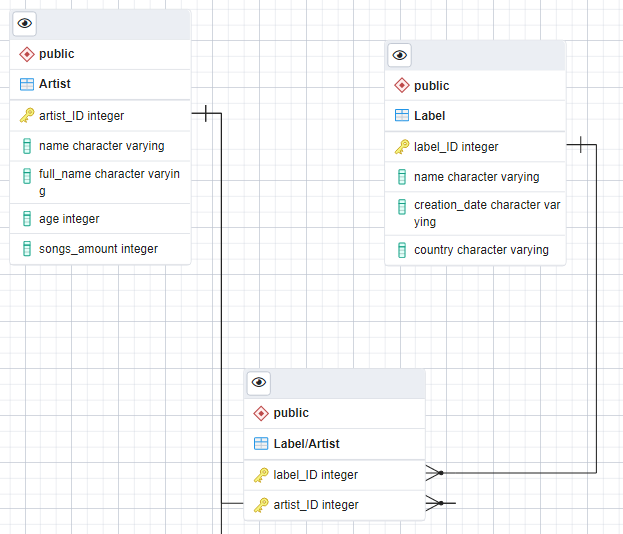
Зв’язок “користувач – пісня” (M:N, адже багато користувачів може слухати багато пісень; тобто цей зв’язок працює як запис та збереження історії прослуховувань, а не лише показує, що зараз слухають користувачі – у такому випадку відношення було б 1:M, адже користувач може одночасно слухати лише одну пісню, але одночасно цю пісню може слухати багато користувачів) зумовив появу додаткової таблиці User/Song, тому що маємо зв’язок багато до багатьох:



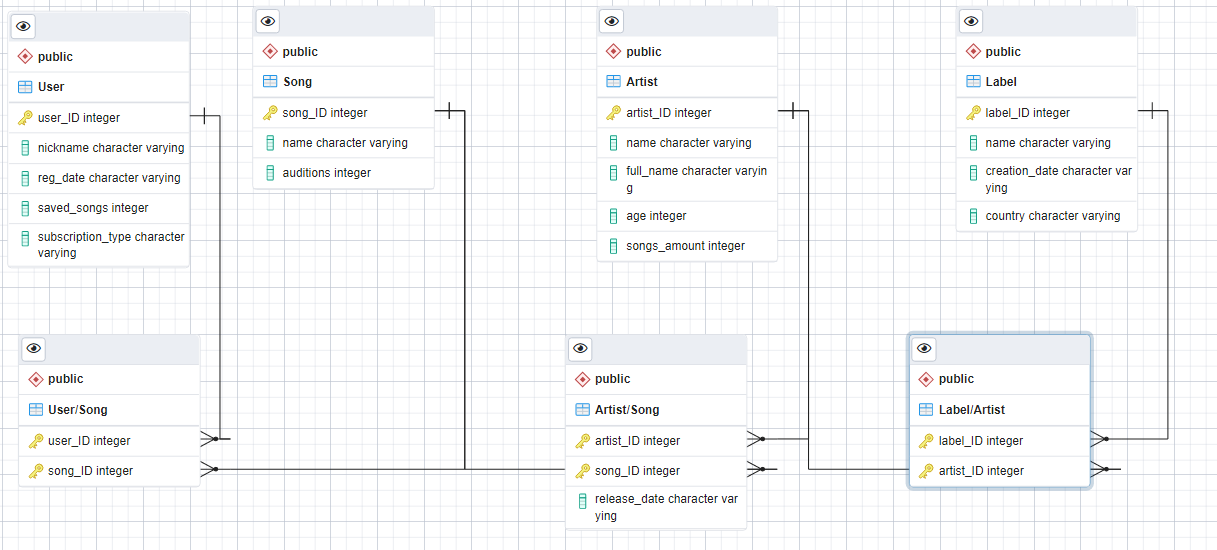
Зв’язок “автор - пісня” (M:N, адже у пісні може бути декілька авторів, як і в одного автора може бути декілька пісень) зумовив появу третьої таблиці Artist/Song (зв’язок має атрибут):



Зв’язок “лейбл-автор” (1:M, адже на одному лейблі може бути багато артистів, але в одного артиста може бути лише один лейбл, або взагалі не бути, тобто опціонально) зумовив появу третьої таблиці Label/Artist:



**Повна схема бази даних**



Значення NOT NULL мають усі ідентифікатори таблиць, окрім ідентифікаторів таблиць User/Song та Label/Artist (пояснюється специфікою зв’язку). Унікальні значення мають всі ідентифікатори таблиць, окрім таблиць зв’язку.

**Пункт №3**

1. **Н.Ф.**:
2. Значення в кожній комірці таблиці мають бути атомарними.

У всіх таблицях створеної бази даних стовпчики не містять множинних атрибутів (повне ім’я автора та різні дати – це один атрибут у нашому випадку, адже реалізуємо просту базу даних, тому не будемо розбивати один атрибут на ще кілька додаткових атрибутів), тобто завжди буде лише одне значення для кожної клітинки. Умова виконується.

1. Кожен рядок має бути унікальним.

Завдяки наборам універсальних ідентифікаторів та їх комбінацій у таблицях зв’язку сутностей, кожен рядок таблиць є універсальним, якщо спеціально не створювати однакових рядків (не має сенсу). Умова виконується.

Виконується обидві умови, отже база даних відповідає 1 нормальній формі.

1. **Н.Ф.:**
2. Задовільнити умови 1 Н.Ф.

Умова виконана.

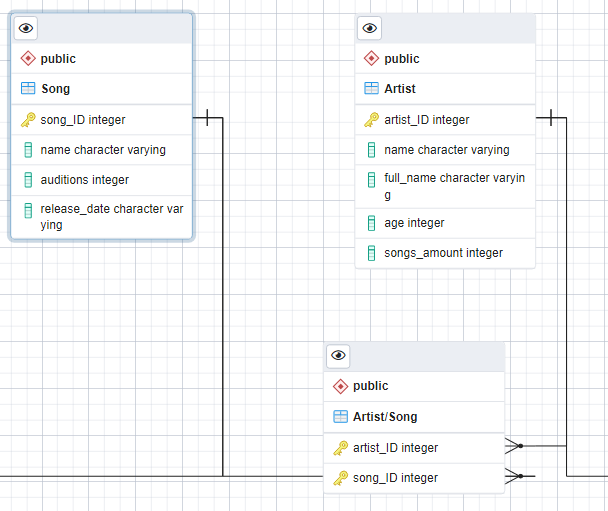
1. Кожен неключовий атрибут має залежати від цілого ключа, а не від його частини.

Більшість таблиць створеної бази даних мають один атрибут у якості первинного ключа, тому для них не перевіряємо цю умову (адже за одним ключовим id атрибутом можливо визначити всі неключові атрибути). Перевіримо для таблиць, де є композитний первинний ключ.

Таблиці User/Song та Label/Artist не мають додаткових атрибутів, тому перевірка непотрібна. Перевіримо таблицю Artist/Song:

artist\_ID, song\_ID release\_date ?

Тут бачимо проблему. Ми можемо визначити дату релізу і без id автора, адже одна й та сама пісня може мати декілька авторів, але дата релізу завжди одна й та сама (звісно пісню можуть перезавантажувати на сервіс, але з точки зору сервісу – це нова композиція). Тому потрібно переробити цей момент. Найкраще рішення, яке я бачу – це прибрати атрибут у цьому зв’язку та додати його до таблиці Song:



Все інше залишаємо без змін.

Перевірку виконано, помилку виправлено.

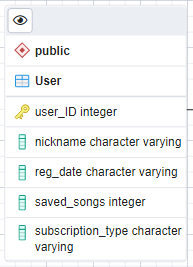
Виконується обидві умови, отже база даних відповідає 2 нормальній формі.

1. **Н.Ф.**:
2. Задовільнити умови 2 Н.Ф.

Умова виконана.

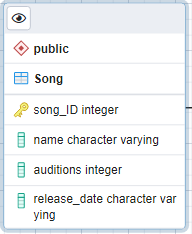
1. Кожен неключовий атрибут не є транзитивнозалежним від кожного кандидатного ключа.

Перевіримо кожну таблицю окремо:



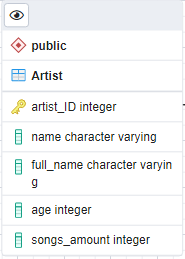
Усі атрибути не є транзитивнозалежними від кожного кандидатного ключа. Питання могли б виникнути при випадку, якби в нас було обмеження на унікальний нікнейм, але в нас його немає, тобто у користувачів можуть бути однакові ніки. Головне – це айді. Тому тут умова виконується:

user\_ID nickname, user\_ID reg\_date, user\_ID saved\_songs, user\_ID subscription\_type



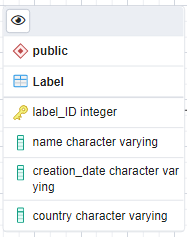
Знову ж таки, усі атрибути не є транзитивнозалежними від кожного кандидатного ключа. Багато пісень можуть мати одну й ту саму назву, тому за цим неможливо визначити кількість прослуховувань та дату релізу. Інші атрибути не є кандидатами на ключ. Тому тут умова виконується:

song\_ID name, song\_ID auditions, song\_ID release\_date

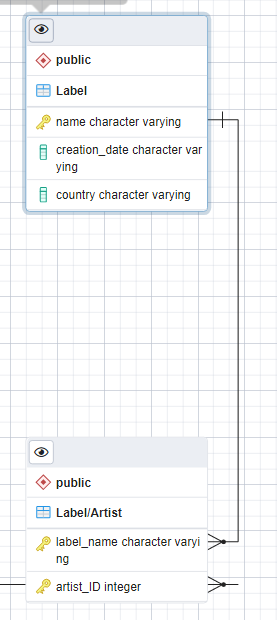


Атрибути name та full\_name не є унікальними. Деякі автори пісень можуть мати однакові сценічні ім’я, або взагалі однакові повні імена (але дуже рідко). Але так як у нас не стоять обмеження на ці атрибути, то за ними хоч і можна інколи визначити інші атрибути, але не завжди, адже ми не можемо бути певні, якщо у деяких артистів однакові ці атрибути. Тому всі атрибути визначаються лише за унікальним атрибутом user\_ID:

artist\_ID name, artist\_ID full\_name, artist\_ID age, artist\_ID songs\_amount. Інші атрибути не є кандидатами на ключ.



На відміну від ситуацій з піснями та артистами, назва лейблу не може бути однаковою у двох лейблах (можуть бути юридичні проблеми). Тому атрибут name може визначати інші атрибути. Потрібно зробити невеличкі зміни:



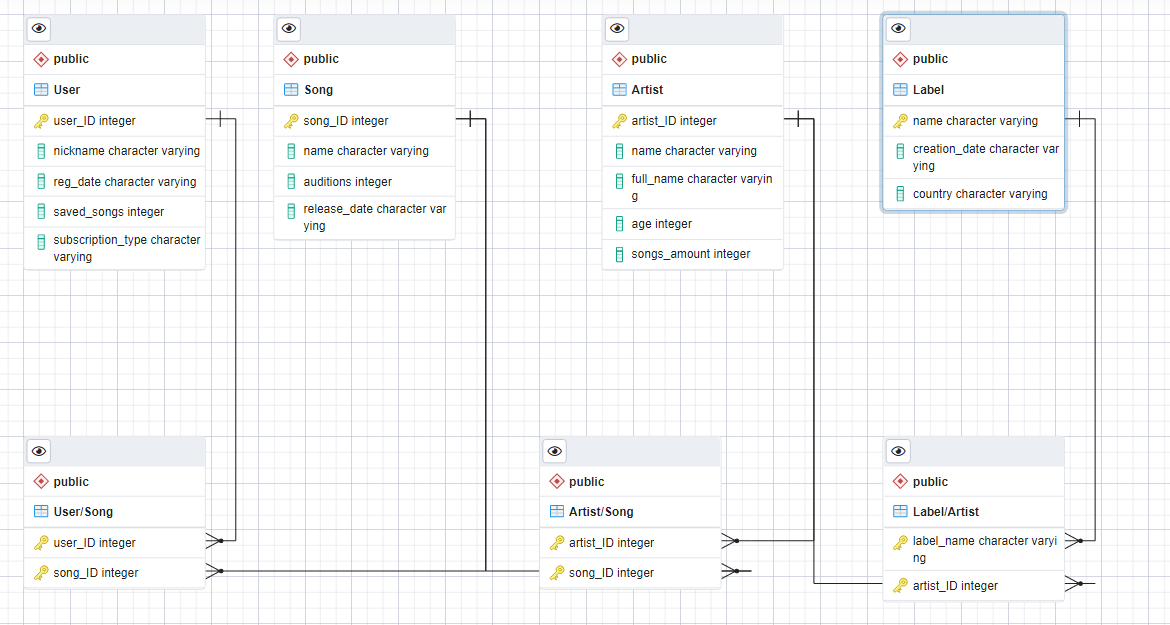
Як бачимо, label\_id нам не потрібне, можемо спростити, видаливши його. PK тепер назва лейблу. Атрибут name – унікальний. Дані зміни не порушують 1 та 2 Н.Ф. і задовольняють 3 Н.Ф.:

name creation\_date, name country.

Таблиці зв’язків можна не розглядати, адже там немає інших атрибутів.

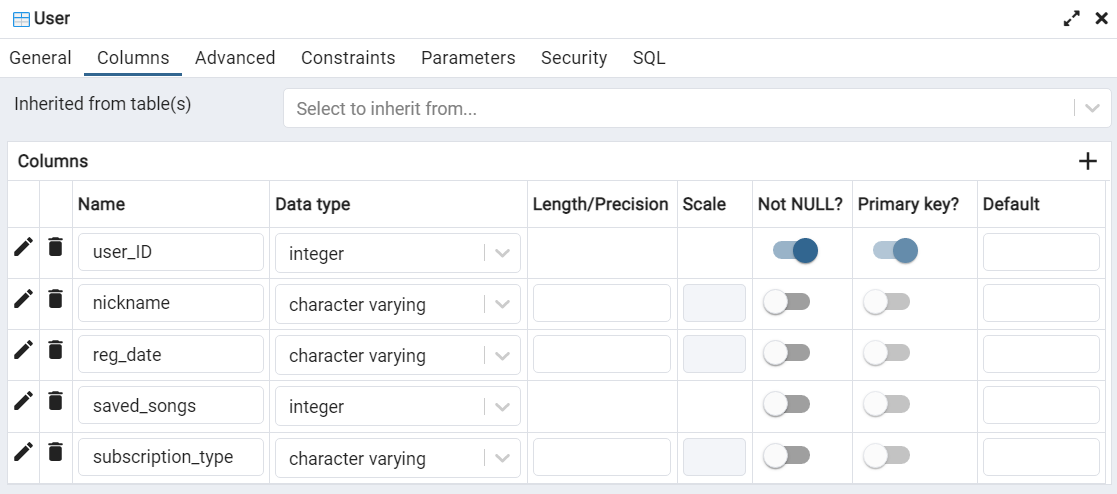
Виконується обидві умови, отже база даних відповідає 3 нормальній формі.

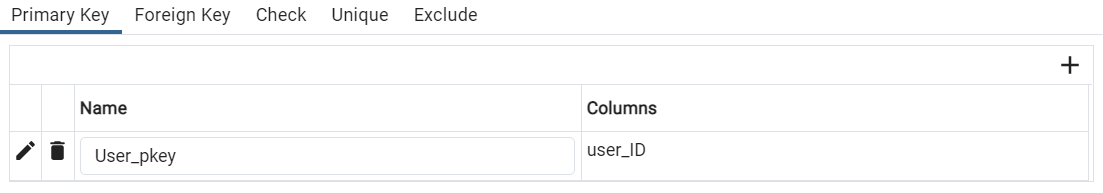
**Оновлена схема**

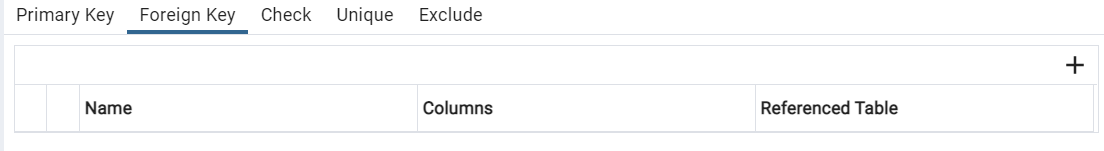


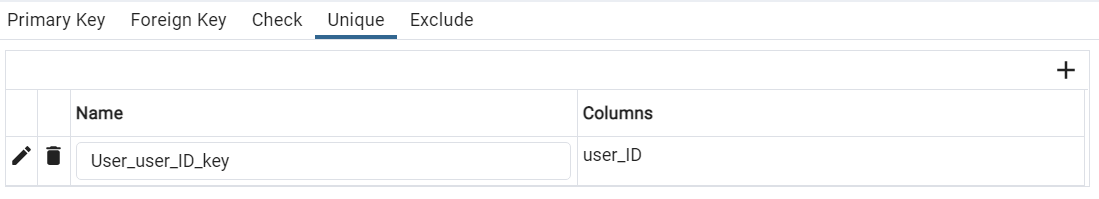
**Пункт №4**

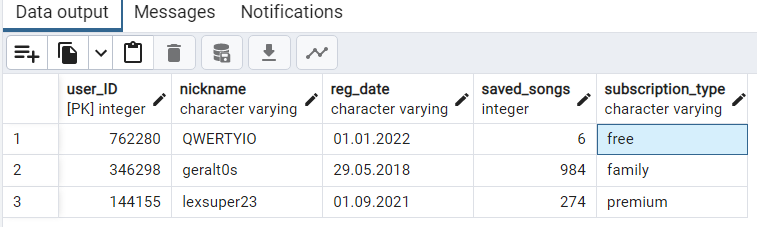
Таблиця “User”



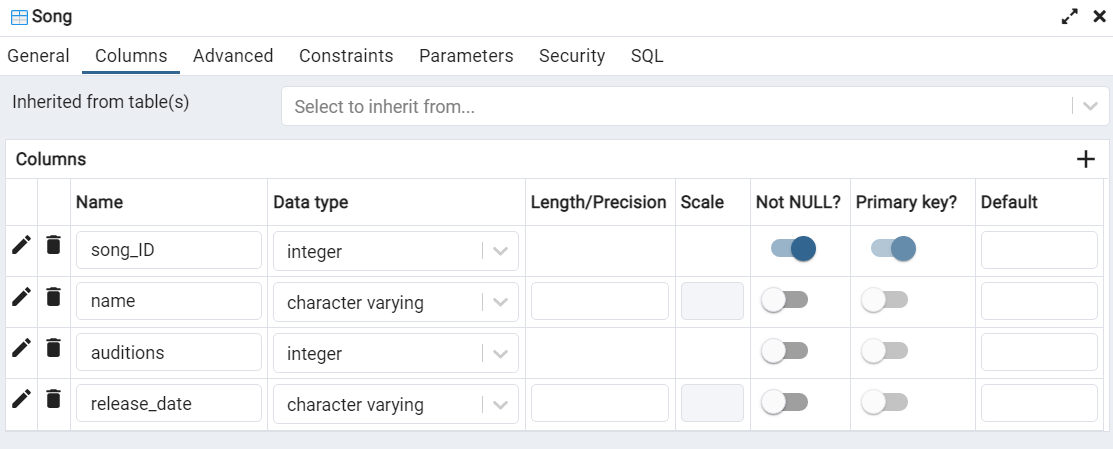


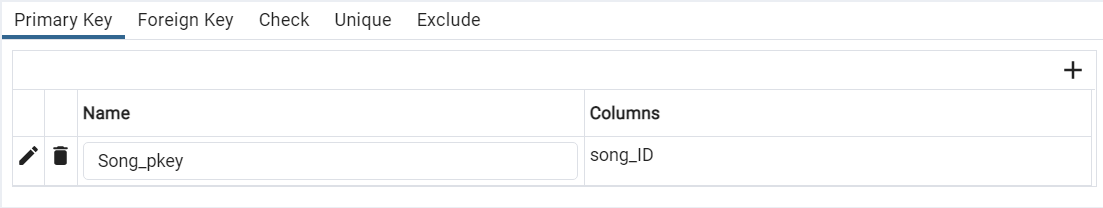


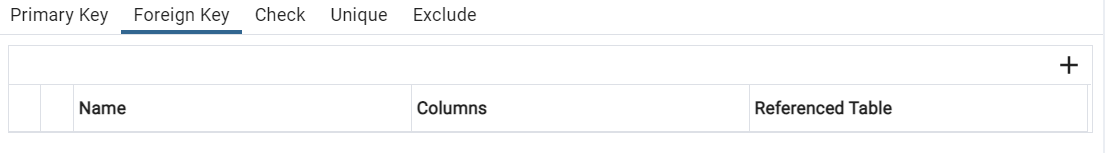


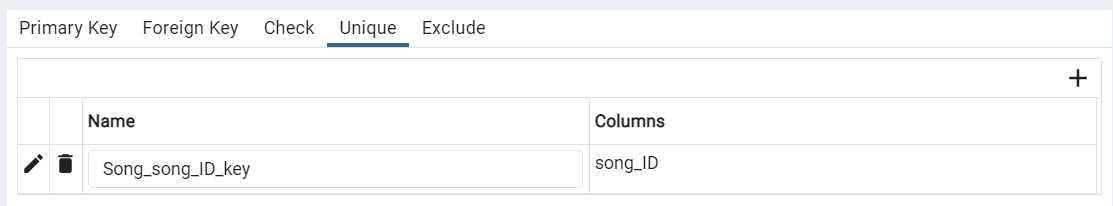


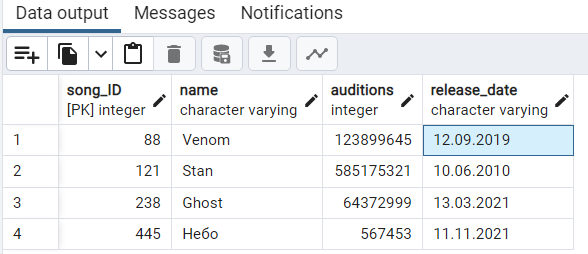
Таблиця “Song”



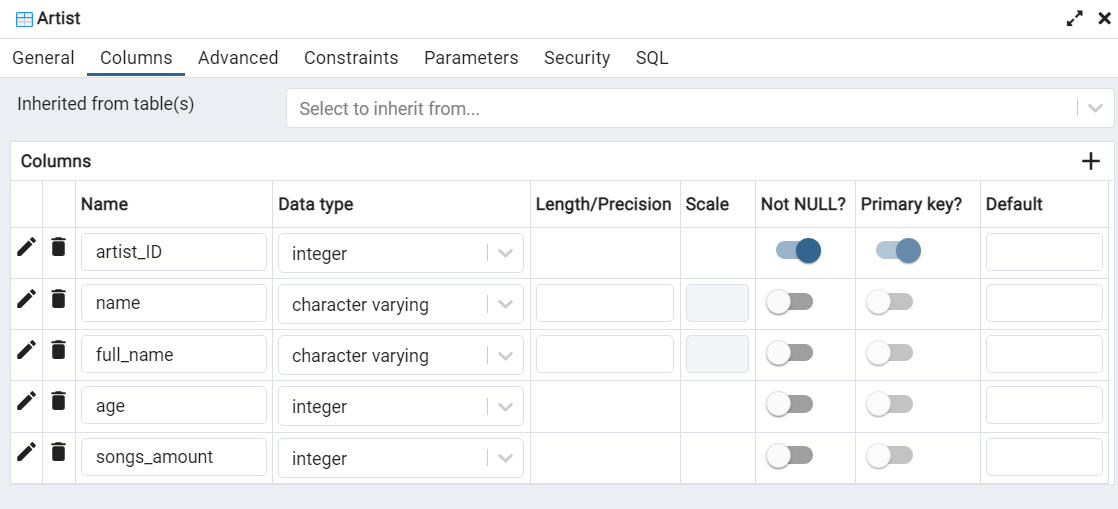


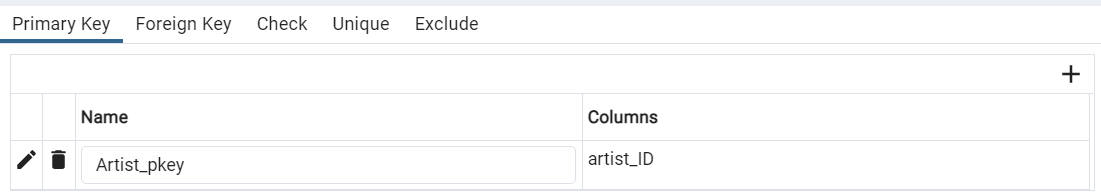


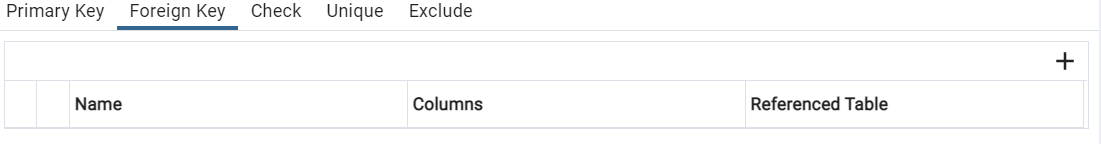


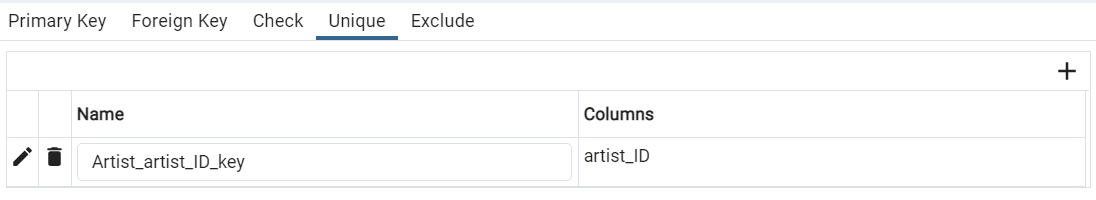


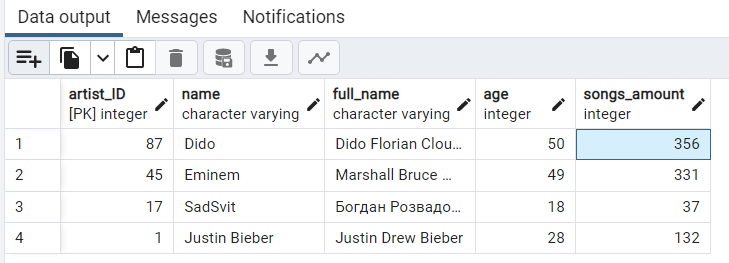
Таблиця “Artist”



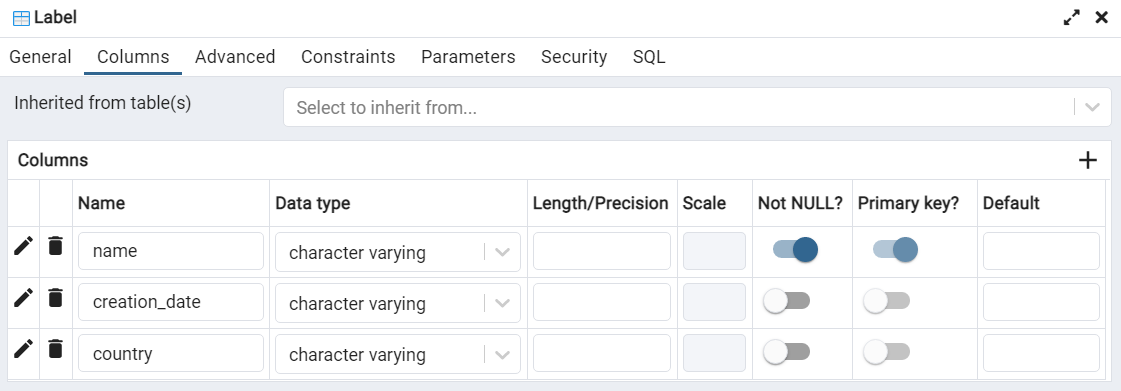


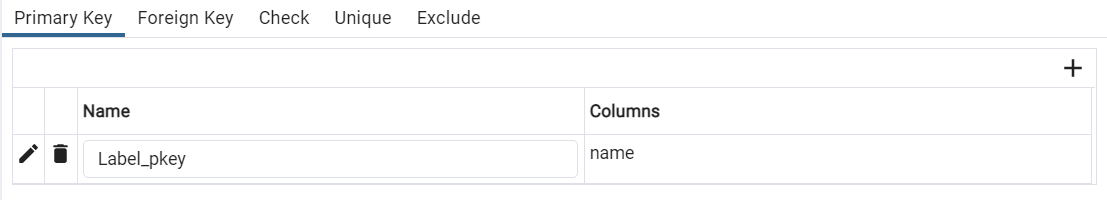


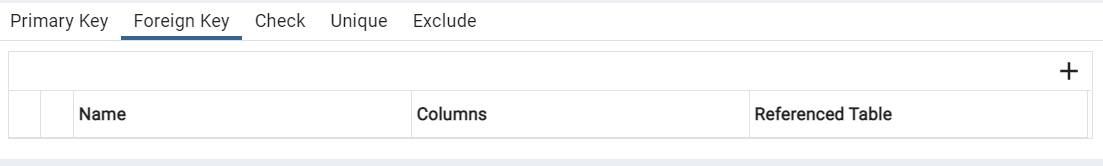


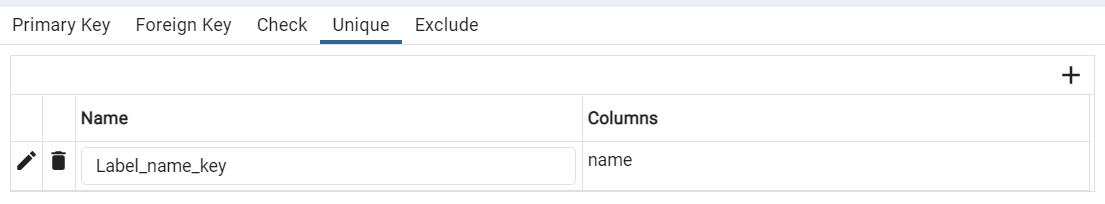


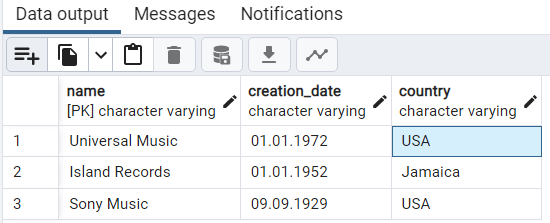
Таблиця “Label”



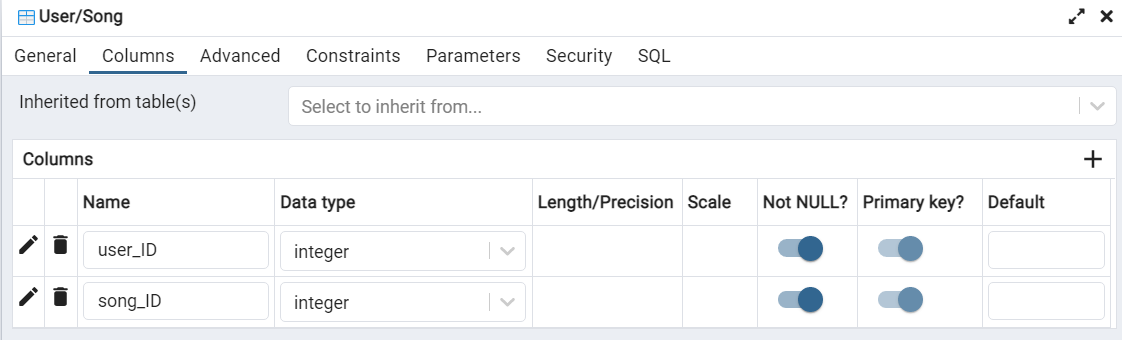


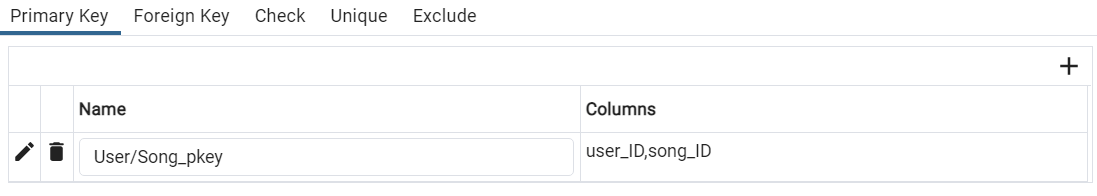


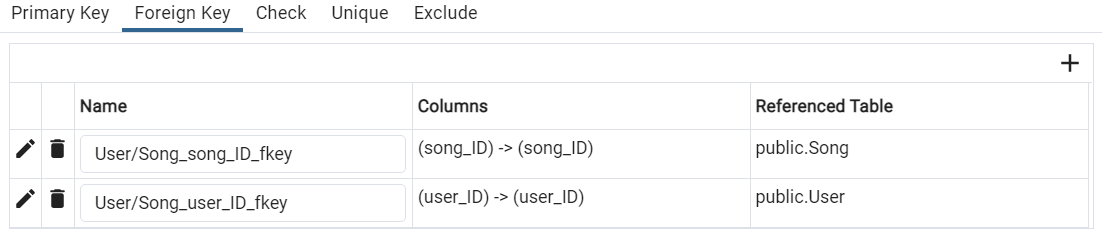
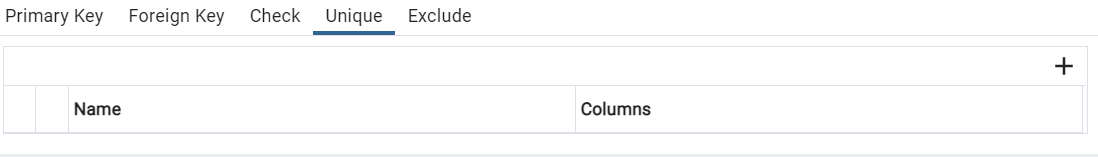


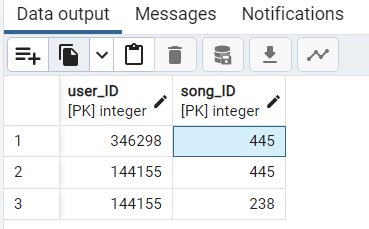


Таблиця “User/Song”

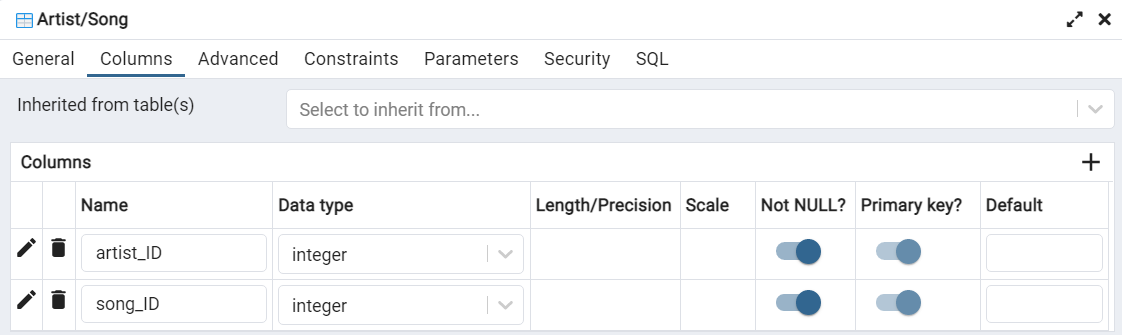




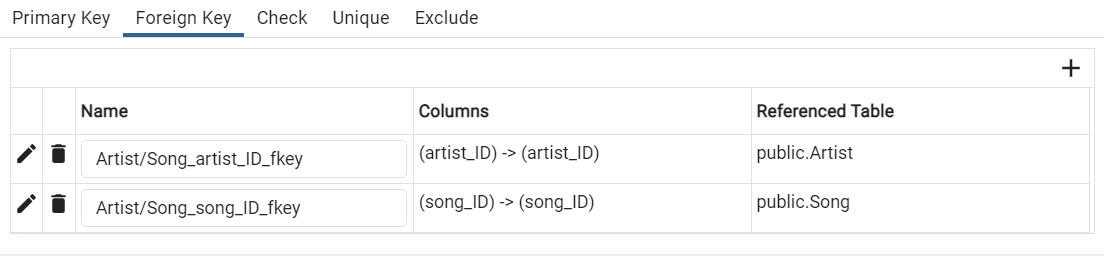
 



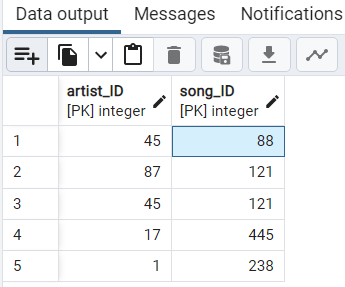
Таблиця “Artist/Song”











Таблиця “Label/Artist”

