

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»**

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

Виконав: студент 3 курсу

ФПМ групи КВ-83

Хаустович Олександр

Перевірив: Павловський В.І.

Київ – 2020

***Мета роботи****:* 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.

***Варіант (предметна область):*** база даних для музичного стрiмiнгового сервісу.

**Звіт**

В музичних сервісах користувачі можуть слухати пісні як окремо, так і в плей листах. Плей листи – це збірки пісень створені користувачами або системою. Пісні можуть знаходитись в декількох плей листах одночасно. Також кожна пісня входить в альбом. Альбом – збірка пісень, яку формує автор пісень, групуючи композиції по жанру/стилю/настрою/ідеї. Пісня може знаходитися лише в одному альбомі. У кожного альбому є один або декілька авторів. У автора може не бути альбомів (якщо автор тільки починає писати пісні) або може бути довільна кількість. У кожного користувача є відповідний рахунок в банку, з якого він щомісяця платить за користування сервісом.

1. В концептуальній моделі предметної області "Музичний стрiмiнговий сервіс" (Рисунок 1) виділяються наступні сутності та зв'язки між ними:

* Сутність «User» з атрибутами: id, name, email, age;
* Сутність «Playlist» з атрибутами: id, name, is\_explicit, creator;
* Сутність «Song» з атрибутами: id, name, is\_explicit;
* Сутність «Album» з атрибутами: id, name, is\_explicit, publication\_date;
* Сутність «Author» з атрибутами: id, name, age, subscribers.

Між сутностями «User» та «Playlist» зв'язок «uses» типу N:M , тому що юзер може використовувати 0 і більше плейлистів, і плейлист можуть використовувати 0 і більше юзерів.

Між сутностями «User» та «Song» зв'язок «listens» типу N:M , тому що юзер може слухати 0 і більше пісень, і пісню можуть слухати 0 і більше юзерів.

Між сутностями «Playlist» та «Song» зв'язок «contains» типу N:M , тому що в плей листі може бути 0 і більше пісень, і пісня може бути в 0 і більше плей листах.

Між сутностями «Song» та «Album» зв'язок «contains» типу 1:N , тому що в альбомі може бути 1 і більше пісень, але пісня може бути лише в одному альбомі.

Між сутностями «Album» та «Author» зв'язок «publishes» типу N:N , тому що у автора може бути 0 і більше альбомів, а у альбому може бути 1 і більше авторів.

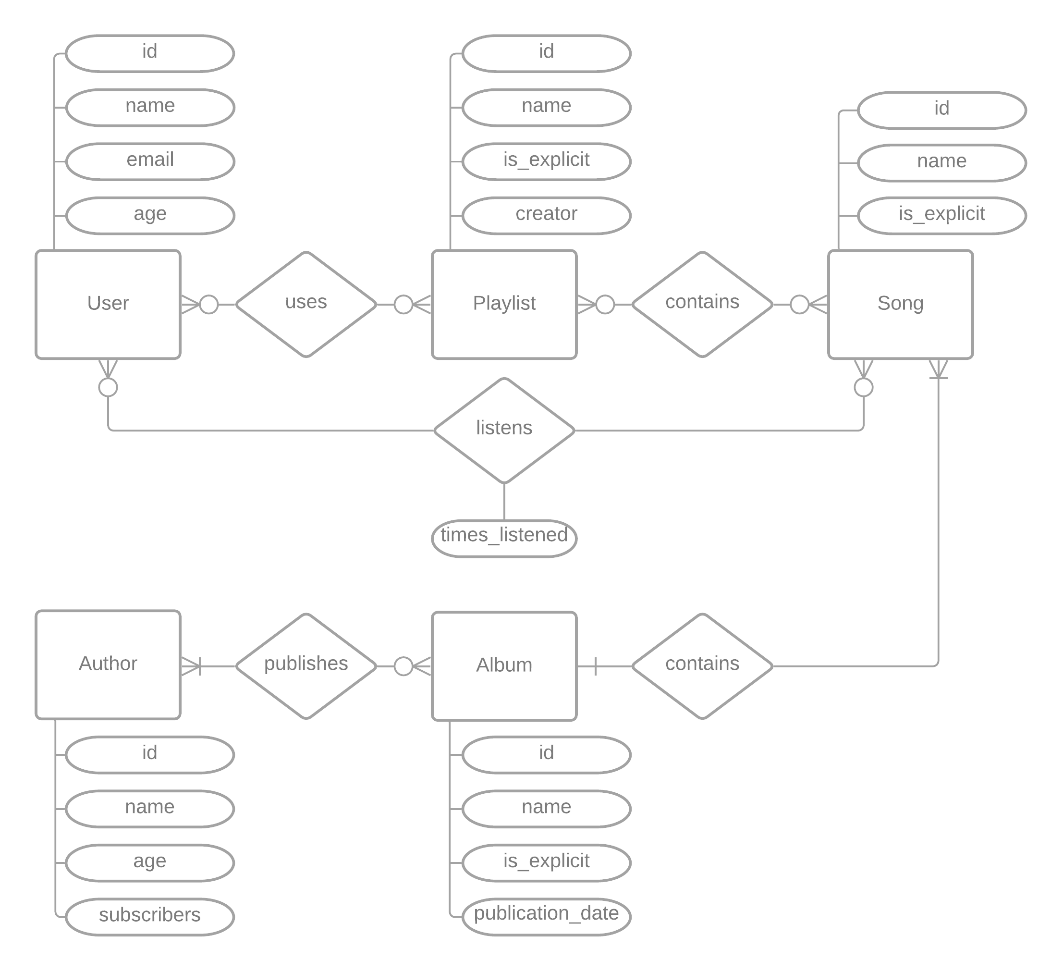
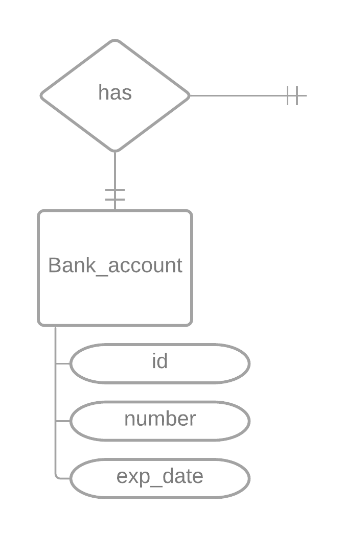
Між сутностями «User» та «Bank\_account» зв'язок «has» типу 1:1 , тому що одному користувачу відповідає один банківський рахунок і навпаки.

Рисунок 1 — ER-модель предметної області "Музичний стрiмiнговий сервіс". Використано сервіс Lucidchart

1. Схема бази даних PostgreSQL на основі ER-моделі предметної області "Музичний стрiмiнговий сервіс" (Рисунок 2).

Сутність «User» було перетворено в таблицю «User».

Сутність «Playlist» було перетворено в таблицю «Playlist».

Зв'язок «uses» типу N:M призвів до появи таблиці «user\_uses\_playlist».

Сутність «Bank\_account» було перетворено в таблицю «Bank\_account».

Зв’язок «has» типу 1:1 призвів до появи в таблиці «Bank\_account» додаткового поля user\_id, що є зовнішнім ключем до поля user\_id таблиці «User» та допускає тільки унікальні значення.

Сутність «Song» було перетворено в таблицю «Song».

Зв'язок «contains» типу призвів до появи таблиці «playlist\_contains\_song».

Зв'язок «listens» типу N:M призвів до появи таблиці «user\_listens\_song»

Сутність «Album» було перетворено в таблицю «Album».

Зв'язок «contains» типу 1:N призвів до появи в таблиці «Song» додаткового поля album\_id.

Сутність «Author» було перетворено в таблицю «Author».

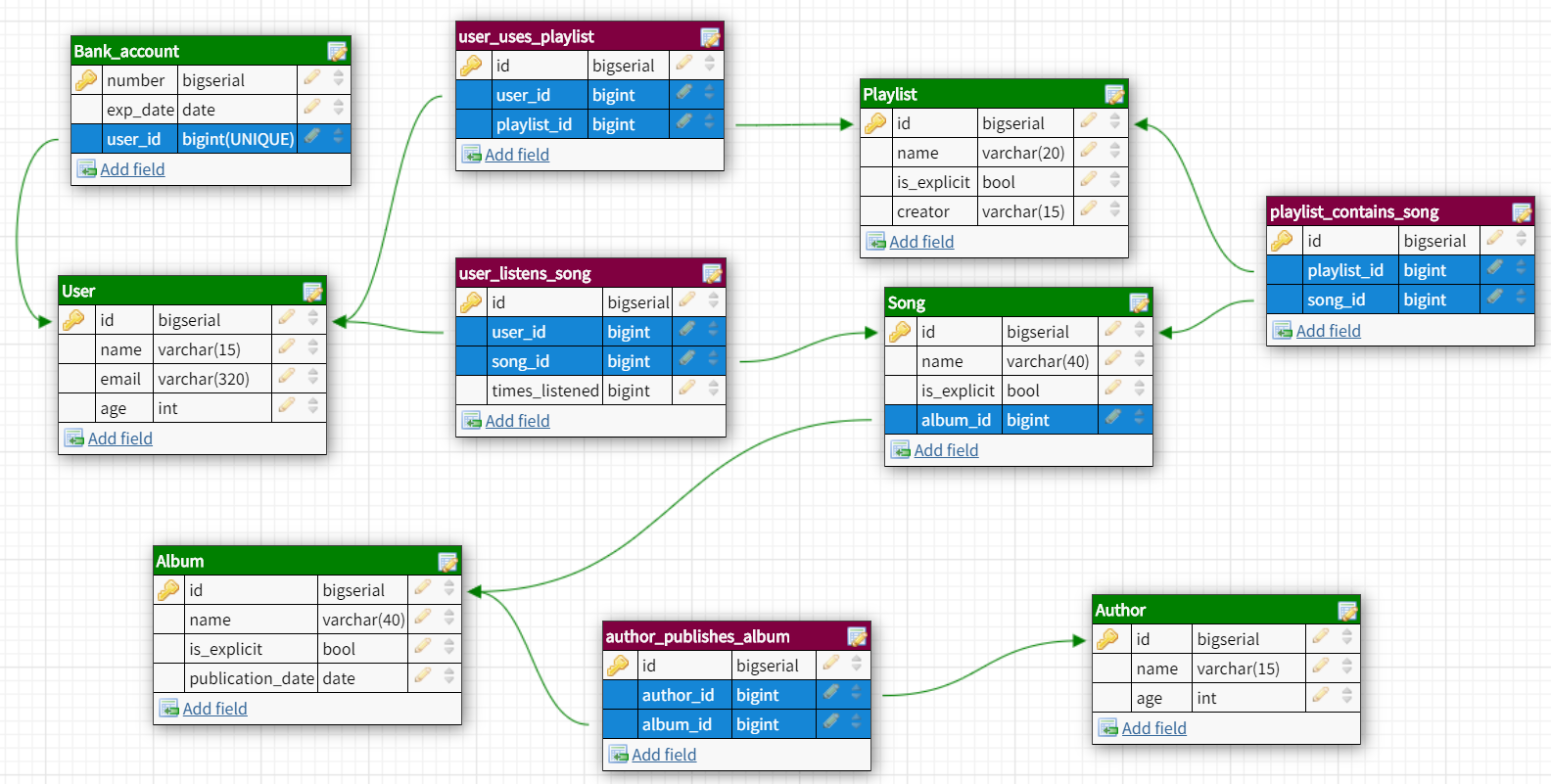
Зв'язок «publishes» типу N:M призвів до появи таблиці «author\_publishes\_album».

Рисунок 2 — Схема бази даних PostgreSQL на основі ER-моделі предметної області "Музичний стрiмiнговий сервіс".

*Примітка*. При побудові схеми БД використано сервіс Dbdesigner

|  |  |  |
| --- | --- | --- |
| **Таблиця** | **Атрибути** | **Тип** |
| User – таблиця для сутності User. Зберігає дані про користувачів | 🔑id – унікальний номер користувача. Не допускає NULL.  name – ім’я користувача. Не допускає NULL.  email – електронна пошта користувача. Не допускає NULL та є унікальним.  age – вік користувача. Не допускає NULL. | bigserial  character varying(15)  character varying(320)  integer |
| Bank\_account – таблиця для сутності Bank\_account. Зберігає дані про банківський рахунок користувача. | 🔑number – унікальний номер банківського рахунку. Не допускає NULL.  exp\_date – дійсний до. Не допускає NULL.  user\_id – унікальний номер власника рахунку. Не допускає NULL та є унікальним. | character varying(10)  date  bigint |
| Playlist – таблиця для сутності Playlist. Зберігає дані про плей листи. | 🔑id – унікальний номер плей листа. Не допускає NULL.  name – ім’я плей листа. Не допускає NULL.  is\_explicit – чи є відвертий контент. Не допускає NULL.  creator-ім’я автора плей листа. | bigserial  character varying(20)  Boolean  character varying(15) |
| Song – таблиця для сутності Song. Зберігає дані про пісні. | 🔑id – унікальний номер пісні  name – назва пісні. Не допускає NULL.  is\_explicit – чи є відвертий контент. Не допускає NULL.  album\_id – id альбому, до якого належить пісня. Не допускає NULL. | bigserial  character varying(40)  boolean  bigint |
| Album – таблиця для сутності Album. Зберігає дані про альбоми. | 🔑id – унікальній номер альбому. Не допускає NULL.  name – назва альбому. Не допускає NULL.  is\_explicit – чи є відвертий контент. Не допускає NULL.  publication\_date – дата публікації альбому. Не допускає NULL. | bigserial  character varying(40)  boolean  date |
| Author – таблиця для сутності Author. Зберігає дані про авторів. | 🔑id – унікальний номер автора  name – ім’я автора. Не допускає NULL.  age – вік автора. | bigserial  character varying(15)  integer |
| user\_uses\_playlist –містить інформаціюпро те, яким плейлистом користувався користувач. | 🔑id – унікальний номер зв’язку. Не допускає NULL.  user\_id – унікальний номер користувача, який використовує плейлист. Не допускає NULL.  playlist\_id – унікальний номер, плей листа, який використовує користувач. Не допускає NULL. | bigserial  big  bigint |
| user\_listens\_song – містить інформацію про те, яку пісню слухав користувач. | 🔑id – унікальний номер зв’язку. Не допускає NULL.  user\_id – унікальний номер користувача, що слухає пісню. Не допускає NULL.  song\_id – унікальний номер пісні, яку слухає користувач. Не допускає NULL.  times\_listened – кількість прослуховувань конкретної пісні конкретним користувачем. За замовчуванням 0. | bigserial  big  bigint  bigint |
| playlist\_contains\_song – містить інформацію про те, в якому альбомі зберігається пісня. | 🔑id – унікальний номер зв’язку. Не допускає NULL.  playlist\_id – унікальний номер плей листа. Не допускає NULL.  song\_id – унікальний номер пісні. Не допускає NULL. | bigserial  bigint  bigint |
| author\_publishes\_album – містить інформацію про те, хто автор(-и) альбому. | 🔑id – унікальний номер зв’язку. Не допускає NULL.  author\_id – унікальний номер автора альбому. Не допускає NULL.  album\_id – унікальний номер альбому . Не допускає NULL. | bigserial  bigint  bigint |

1. Нормалізація БД до третьої нормальної форми

У всіх таблицях бази даних атрибути зберігають лише одне значення та кожен запис унікальний, тому база даних вже нормалізована до першої форми.

Оскільки у таблицях БД не використовуються композитні ключі, то база даних нормалізована до другої форми.

Перевіримо всі таблиці на наявність транзитивних зв’язків:

1. «User»:

* id → name, email, age;
* id → name;
* id → email;
* id → age;
* name ↮ email;
* email ↮ age;
* age ↮ name.

1. «user\_uses\_playlist»:

* id → user\_id, playlist\_id;
* id → user\_id;
* id → playlist\_id;
* user\_id ↮ playlist\_id.

1. «Playlist»:

* id → name, is\_explicit, creator;
* id → is\_explicit;
* id → creator;
* name ↮ is\_explicit;
* is\_explicit ↮ creator;
* creator ↮ name.

1. «Song»:

* id → name, is\_explicit, album\_id;
* id → is\_explicit;
* id → album\_id;
* id → name;
* name ↮ is\_explicit;
* is\_explicit ↮ album\_id;
* album\_id ↮ name.

1. «user\_listens\_song»:

* id → user\_id, song\_id, times\_listened;
* id → user\_id;
* id → song\_id;
* id → times\_listened;
* user\_id ↮ playlist\_id;
* playlist\_id ↮ times\_listened;
* times\_listened↮ user\_id.

1. «playlist\_contains\_song»:

* id → song\_id, playlist\_id
* id → song\_id
* id → playlist\_id
* song\_id ↮ playlist\_id

1. «Album»:

* id → name, is\_explicit, publication\_date
* id → is\_explicit
* id → publication\_date
* id → name
* name ↮ is\_explicit
* is\_explicit ↮ publication\_date
* publication\_date ↮ name

1. «author\_publishes\_album»:

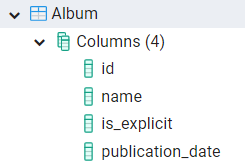
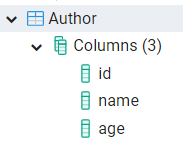
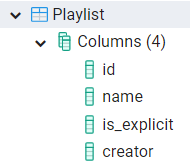
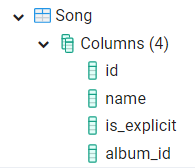
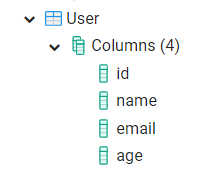
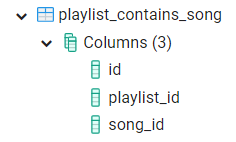
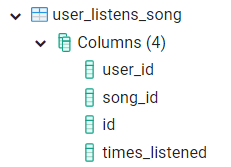
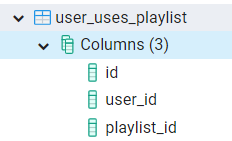
* id → album\_id, author\_id
* id → album\_id
* id → author\_id
* album\_id ↮ author\_id

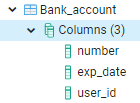
1. «Author»:

* id → name, age
* id → name
* id → age
* name ↮ age

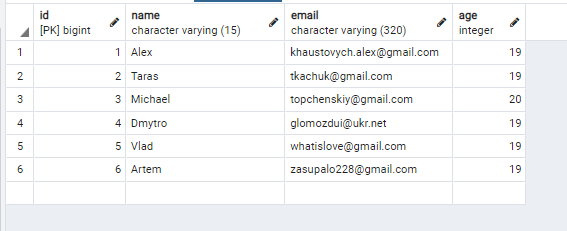
В таблицях немає транзитивних зв’язків – отже вона нормалізована до третьої форми.

1. Створення БД в PostgreSQL з використанням pgAdmin4

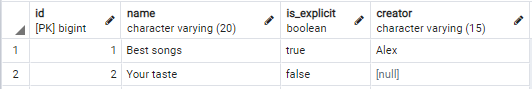
Структура БД Музичний стрiмiнговий сервіс в pgAdmin4        



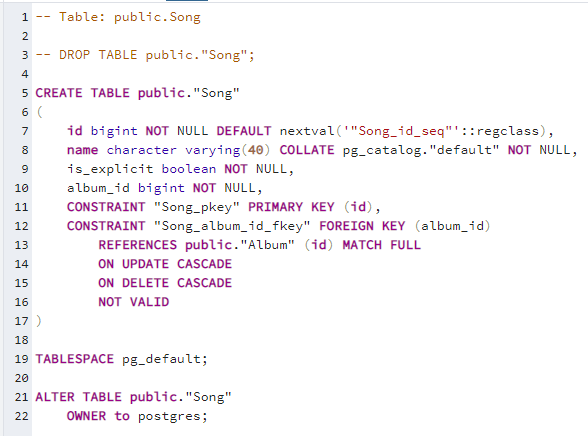
User:



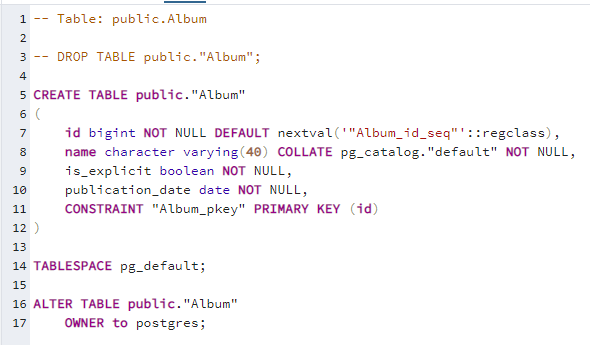
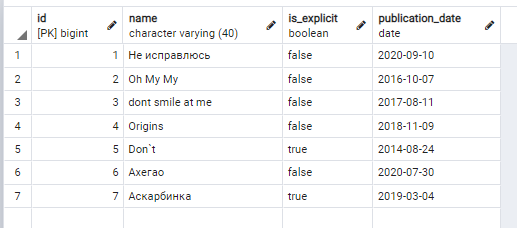
Playlist:



Song:



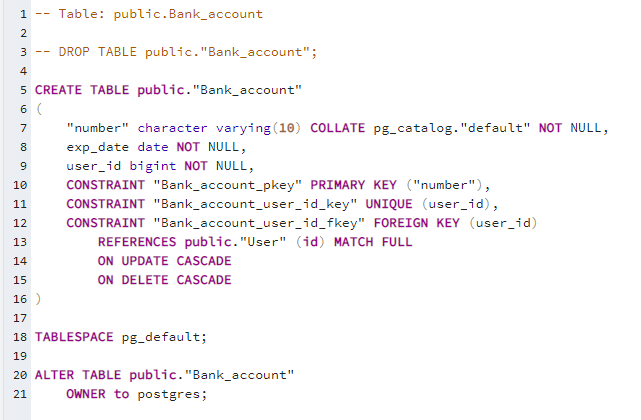
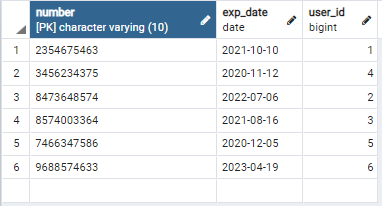
Album:



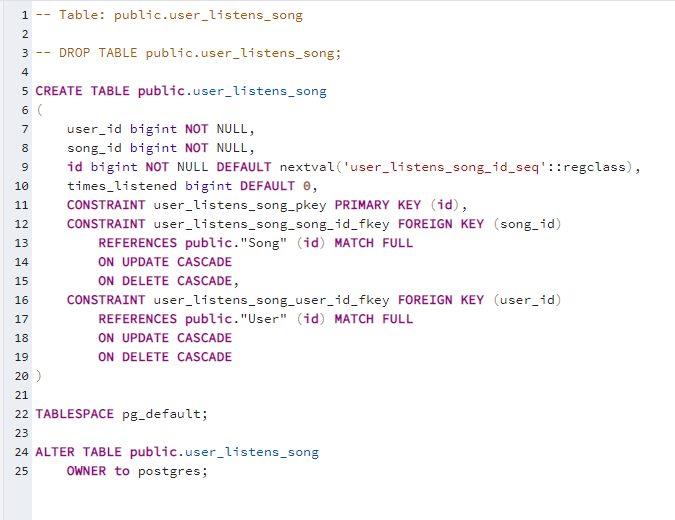
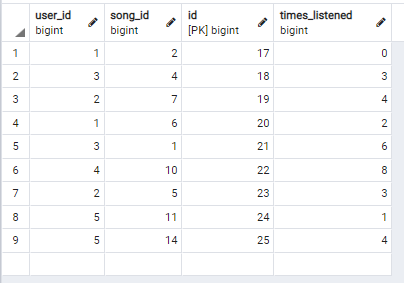
Author:



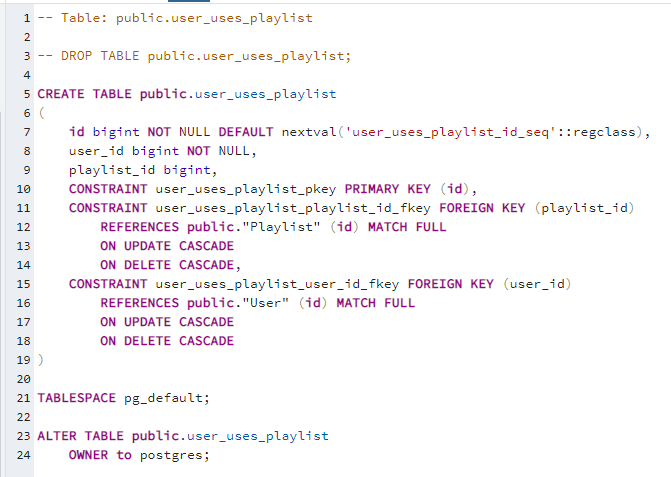
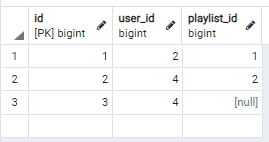
Bank\_account:



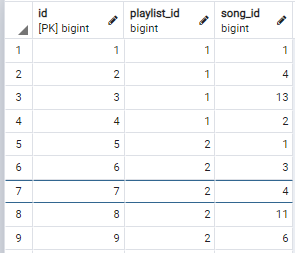
user\_listens\_song:



user\_uses\_playlist:



playlist\_contains\_song:



author\_publishes\_album:

