

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 та засоби управління»**

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

Виконала:

студентка ІII курсу

групи КВ-02

Дахал К.

Перевірив:

доц. Павловський В.І.

Київ 2022

[*Посилання на репозиторій git.hub*](https://github.com/arrooww/bd_lab1)*.*

*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.

**Опис предметної області «Музичний гурт»**

Обрана предметна область передбачає створення гуртом музичних альбомів та пісень. Кожна група має свій альбом. Один альбом включає в себе декілька пісень. Кожен гурт може мати декілька студій звукозапису. Одна студія звукозапису може мати декілька гуртів.

**Сутності предметної області**

Для побудови бази даних для обраної області було виділено сутності, зображені на рисунку 1:

1. Гурт (Band), з атрибутами: код гурту, імʼя, учасники, дата створення. Описує базові дані та склад певного гурту. Кожен гурт має альбоми та пісні.
2. Альбом (Album), з атрибутами: код альбому, код гурту, назва, дата випуску, пісні, тривалість. Описує ключові дані та пісні певного альбому. Кожен альбом має одного виконавця та містить декілька пісень.
3. Пісня (Song), з атрибутами: код пісні, код альбому, код гурту, назва, тривалість. Опис певної пісні. Кожна пісня має одного виконавця та походить з одного альбому.
4. Студія звукозапису (Recording studio), з атрибутами: назва, код студії, код гурту.

**Опис зв’язків між сутностями предметної області**

Сутність “Гурт” має зв’язок 1:N по відношенню до сутності “Альбом”, тому що один гурт може мати декілька альбомів і ці альбоми належатимуть лише одному гурту.

Також сутність “Гурт” має зв’язок 1:N до сутності “Пісня”, тому що один гурт створює декілька пісень і відповідно всі ці пісні мають одного виконавця.

Сутність “Альбом” має зв’язок 1:N до сутності “Пісня”, тому що один альбом має декілька пісень, у свою чергу всі ці пісні можуть належати лише до одного альбому.

Сутність “Студія звукозапису” має звʼязок M:N до сутності “Гурт”, тому що кожен гурт може мати декілька студій звукозапису, а одна студія звукозапису може записувати декілька гуртів.

**Концептуальна модель предметної області “Музична група”**

Концептуальна модель наведена на рисунку 1.

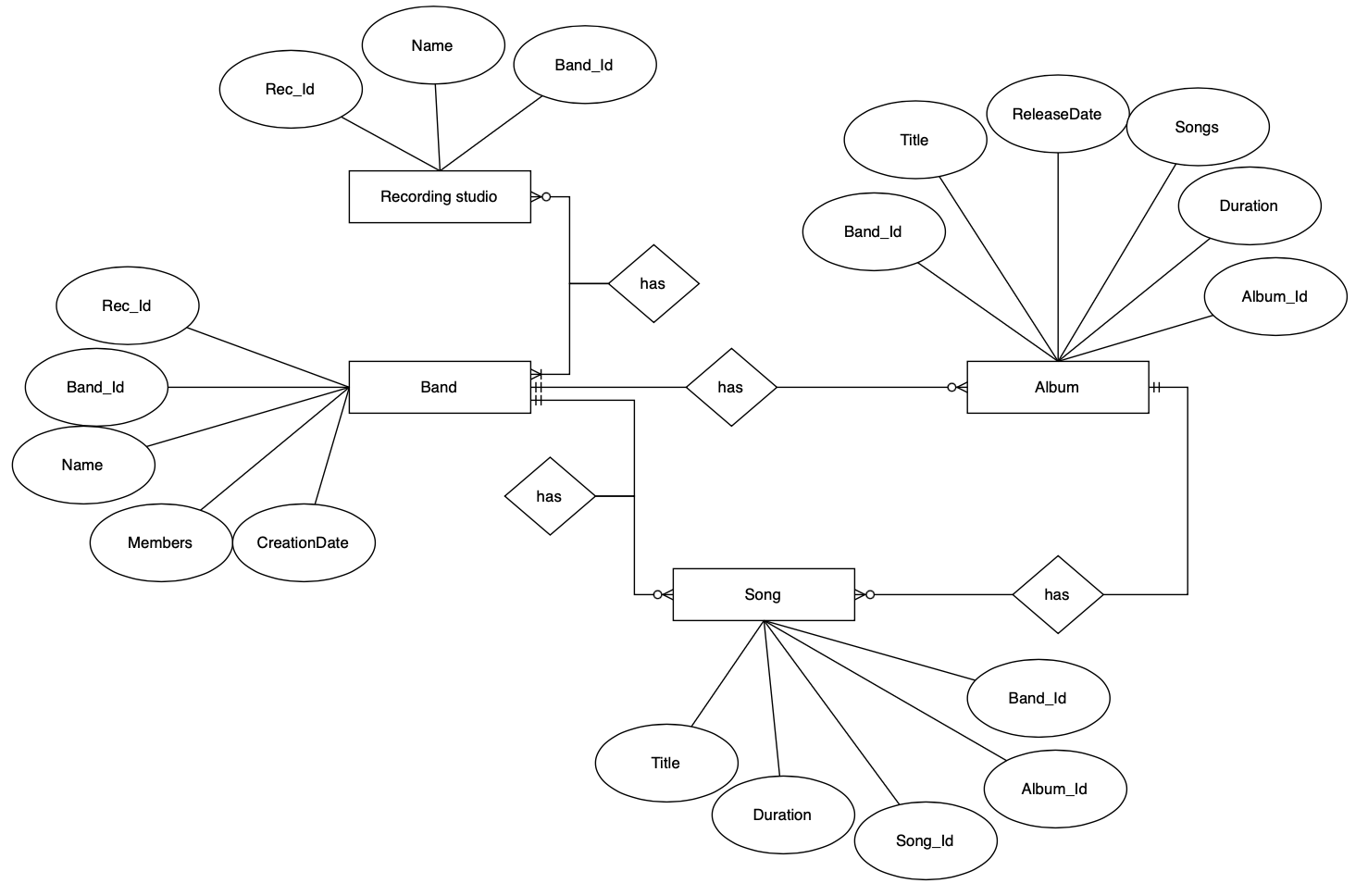


Рисунок 1– ER-діаграма за нотацією “Crow’s foot”.

**Перетворення концептуальної моделі в логічну схему бази даних**

Сутність “Гурт” перетворено у таблицю “Band”. Rec\_Id зберігає зовнішній ключ для звʼязку з таблицею Recording studio.

Сутність “Альбом” перетворено у таблицю “Album”. Band\_Id зберігає зовнішній ключ для звʼязку з таблицею Band.

Сутність “Пісня” перетворено у таблицю “Song”. Band\_Id зберігає зовнішній ключ для звʼязку з таблицею Band. Album\_Id зберігає зовнішній ключ для звʼязку з таблицею Album.

Сутність “Студія звукозапису” перетворено у таблицю “Recording studio”. Для реалізації M:N було використано проміжну таблицю band\_rec з 2-ма зовнішніми ключами, які звʼязані з первинними ключами вихідних таблиць. Band\_Id зберігає зовнішній ключ для звʼязку з таблицею Band.

**Логічна модель (схема) БД «Музичний гурт»**

Логічну модель (схему бази даних) наведено на рисунку 2.

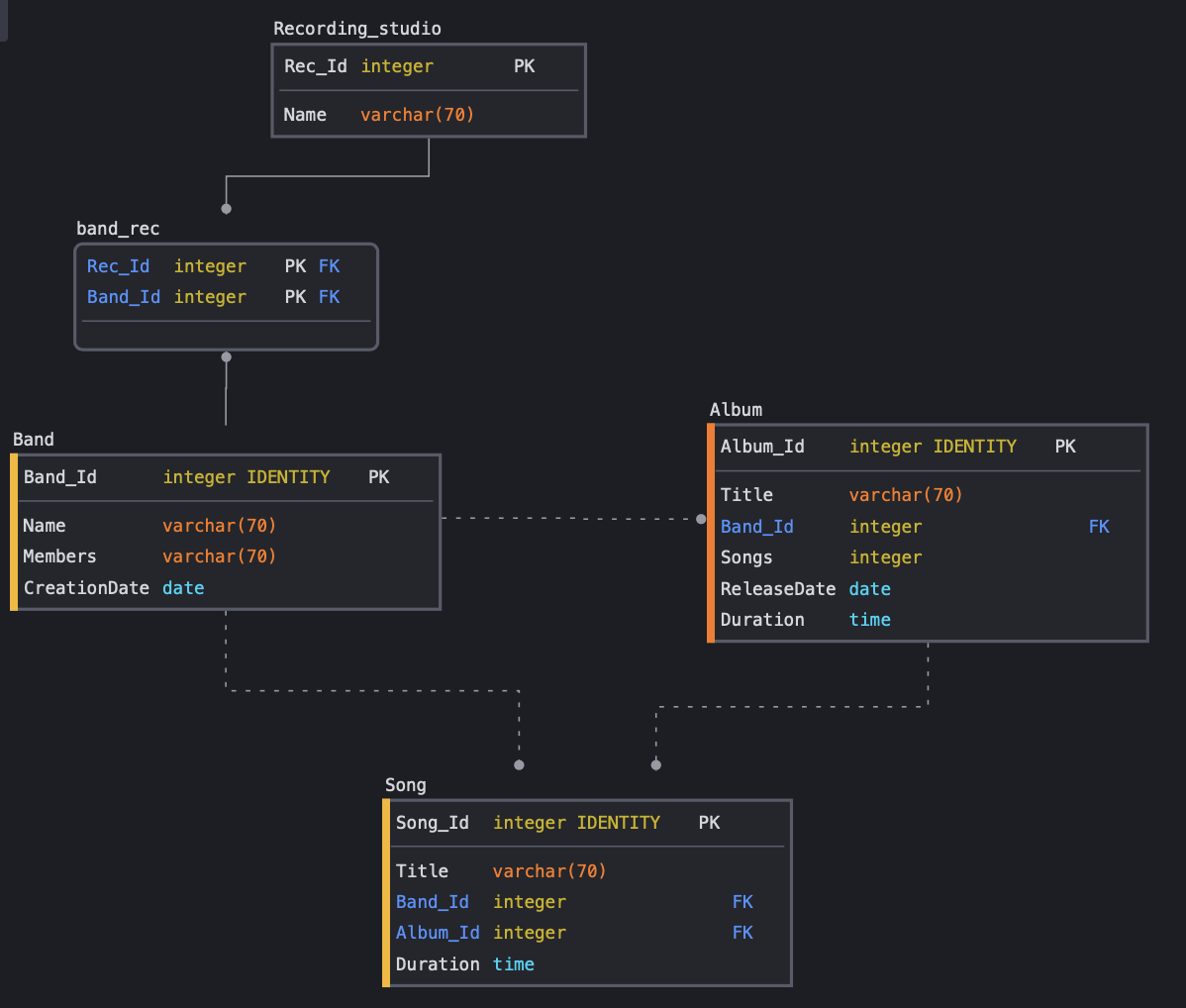


Рисунок 2 – Схема бази даних (інструмент: sqldbm.com)

**Опис об’єктів бази даних у вигляді таблиці**

Опис сутностей та їх атрибутів наведено в таблиці 1.

Таблиця 1 – Опис структури бази даних «Музичний гурт».

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| **Band** – *описує базові дані та склад певного гурту.* | **Band\_Id***- унікальний ідентифікатор гурту.*  **Name** *- назва гурту*.  **Members** - *учасники гурту.*  **CreationDate** *- дата випуску альбому.* | integer *(числовий)*  character varying *(рядок)*  character varying *(рядок)*  data *(дата)* |
| **Album** – *описує ключові дані та пісні певного альбому.* | **Album** \_**Id** *- унікальний ідентифікатор альбому.*  **Band\_Id** *- унікальний ідентифікатор гурту.*  **Title** *– назва альбому.*  **Songs** *- кількість пісень.*  **ReleaseDate** *- дата випуску альбому.*  **Duration** *- тривалість.* | integer *(числовий)*  integer *(числовий)*  character varying *(рядок)*  integer *(числовий)*  data *(дата)*  *time (час)* |
| **Song** - *опис певної пісні.* | **Song\_Id** *- унікальний ідентифікатор пісні.*  **Title** *- назва пісні.*  **Album\_Id** - *унікальний ідентифікатор альбому.*  **Band\_Id** - *унікальний ідентифікатор гурту.*  **Duration -** тривалість пісні. | integer *(числовий)*  character varying *(рядок)*  integer *(числовий)*  integer *(числовий)*  time *(час)* |
| **Recording\_studio** – *описує певну студію звукозапису.* | **Rec\_Id** - *унікальний ідентифікатор студії.*  **Name -** *назва гурту*. | integer *(числовий)*  character varying *(рядок)* |
| **Band\_rec** – *зберігає зовнішні ключі таблиць Recording\_studio та Band.* | **Band\_Id** - *унікальний ідентифікатор гурту.*  **Rec\_Id** - *унікальний ідентифікатор студії.* | integer *(числовий)*  integer *(числовий)* |

**Функціональні залежності для кожної таблиці**

Band:

Band\_Id → Name, Members, CreationDate

Band\_Id → Name (назва залежить від коду гурта)

Band\_Id → Members (учасники залежать від коду гурта)

Band\_Id → CreationDate (дата створення залежить від коду гурта)

Album:

Album\_Id → Title, Songs, ReleaseDate, Duration, Band\_Id

Album\_Id → Title (назва залежить від коду альбому)

Album\_Id → Songs (кількість пісень залежить від коду альбому)

Album\_Id → ReleaseDate (час випуску залежить від коду альбому)

Album\_Id → Duration (тривалість залежить від коду альбому)

Album\_Id → Band\_Id (код гурту залежить від коду альбому)

Song:

Song\_Id → Title, Duration, Album\_Id, Band\_Id

Song\_Id → Title (назва залежить від коду пісні)

Song\_Id → Duration (тривалість залежить від коду пісні)

Song\_Id → Album\_Id (код альбому залежить від коду пісні)

Song\_Id → Band\_Id (код гурту залежить від коду пісні)

Recording\_studio:

Rec\_Id → Name

Rec\_Id → Name (назва залежить від коду студії)

**Відповідність схеми бази даних до третьої нормальної форми**

Схема відповідає 1НФ, тому що:

1. В таблиці немає дубльованих рядків.
2. В кожному стовпці зберігаються дані одного типу.
3. В кожній комірці зберігається скалярне значення.

Схема відповідає 2НФ, тому що:

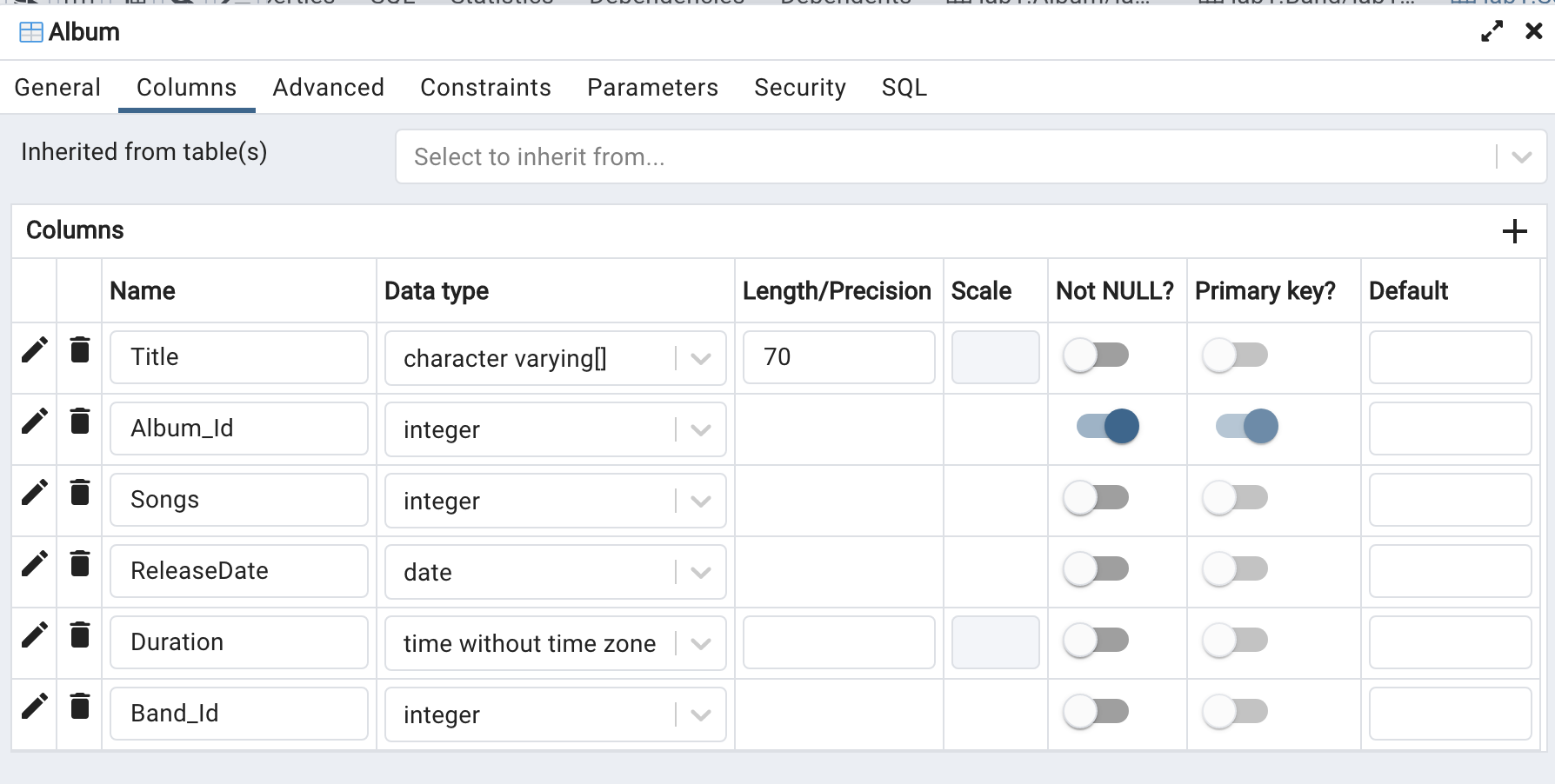
1. Вона відповідає 1НФ.
2. Має первинний ключ, а всі не ключові стовпці таблиці залежать від первинного ключа.

Схема відповідає 3НФ, тому що:

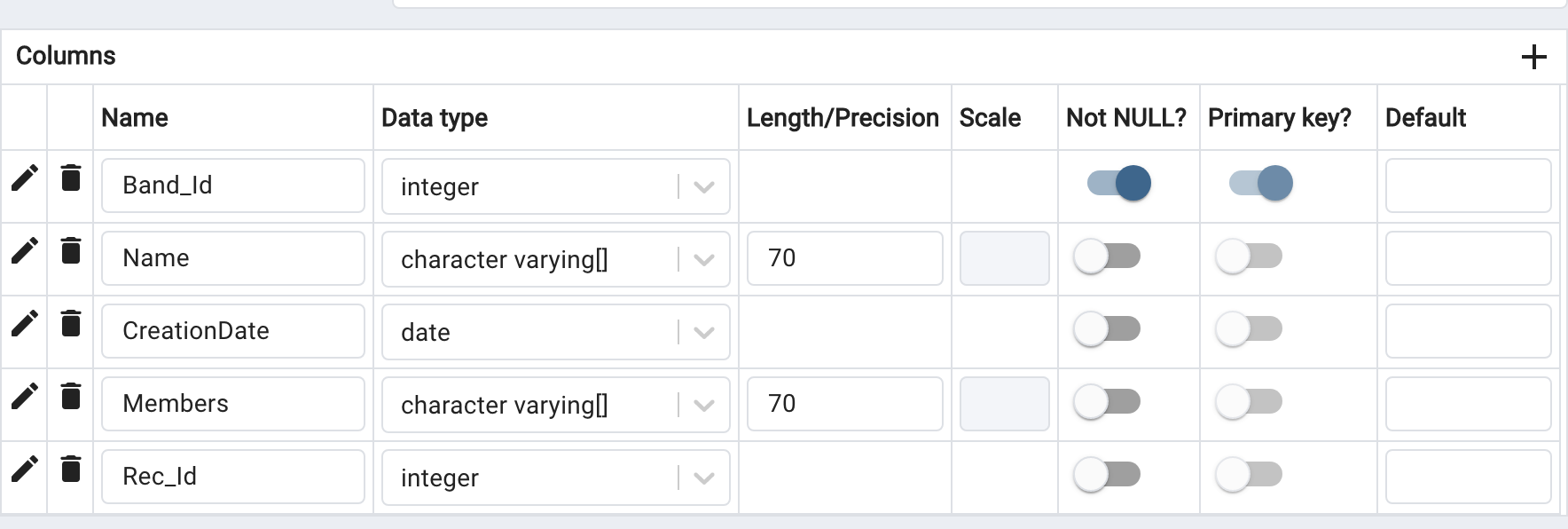
1. Вона відповідає 2НФ.
2. Кожен неключовий атрибут безпосередньо залежить від первинного ключа.

**Фізична модель БД «Музичний гурт» у pgAdmin4**

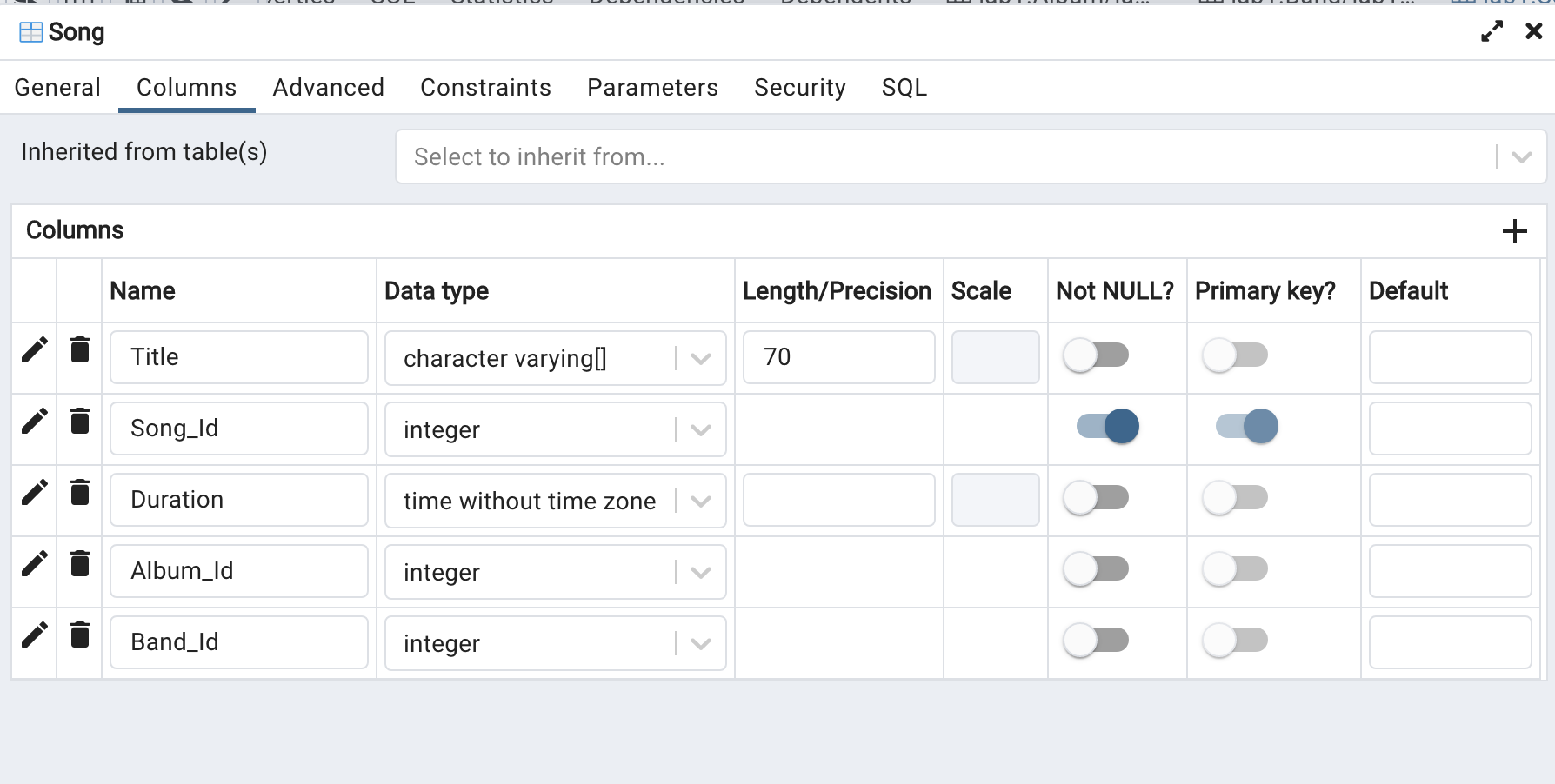
**Колонки таблиці “Album”:**

****

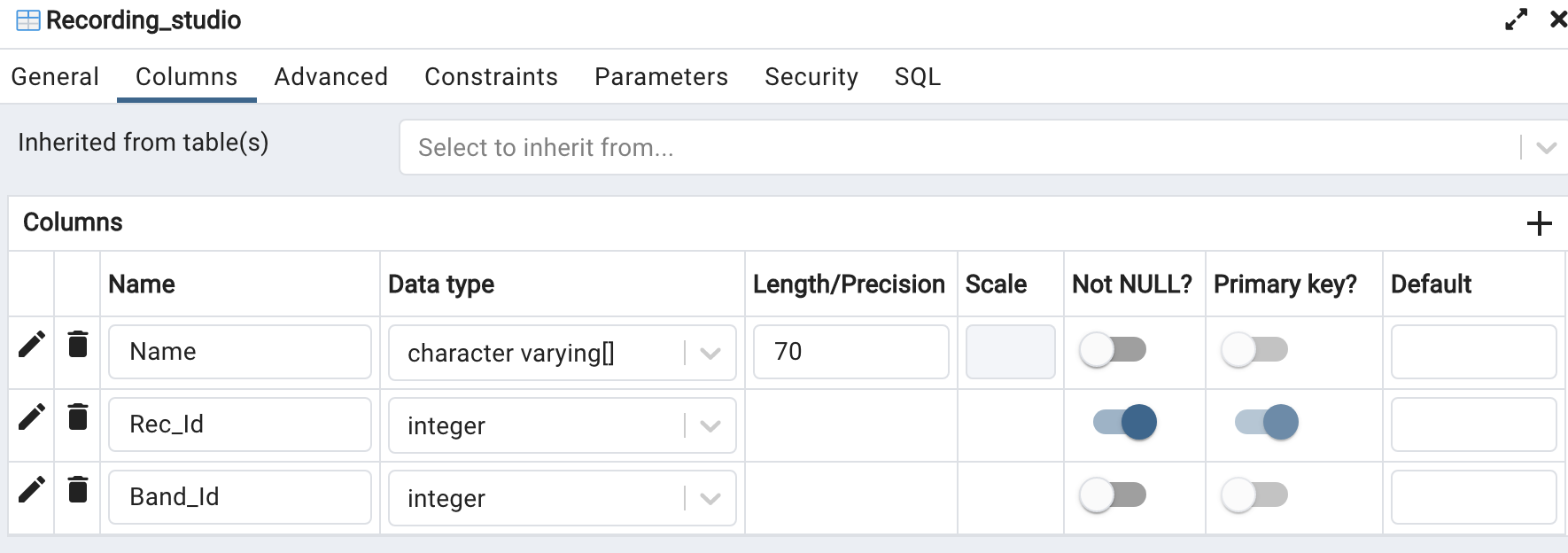
**Колонки таблиці “Band”:**

****

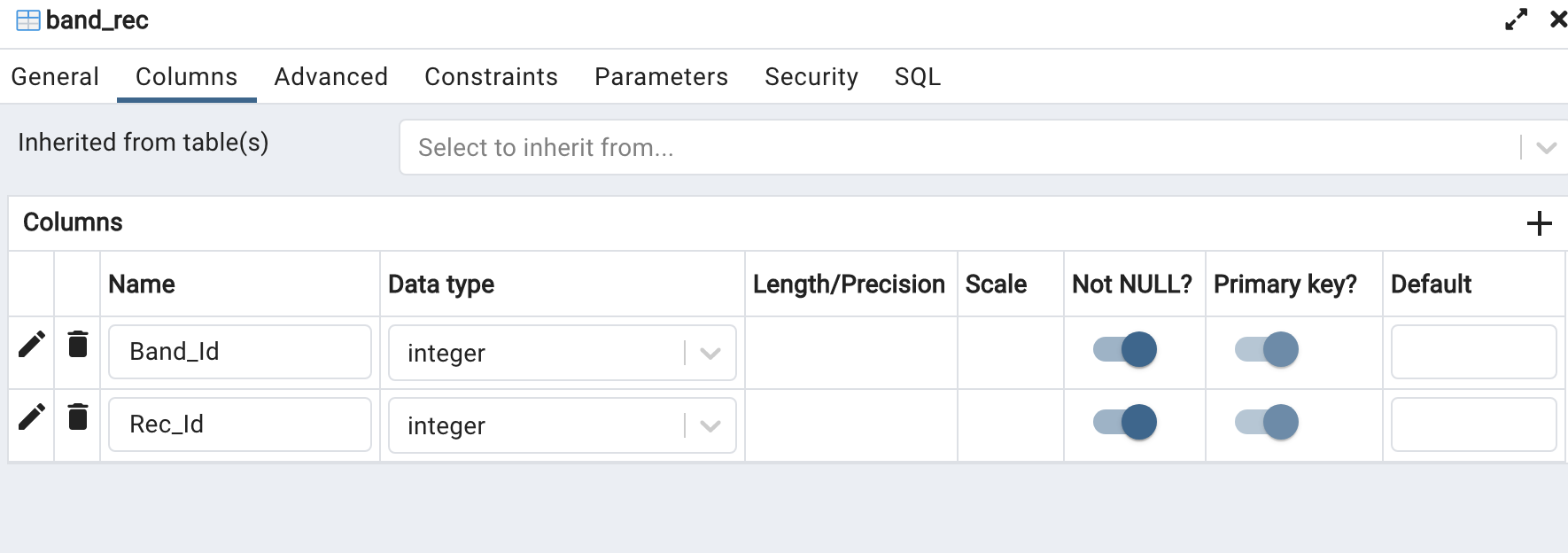
**Колонки таблиці “Song”:**

****

**Колонки таблиці “Recording\_studio”:**

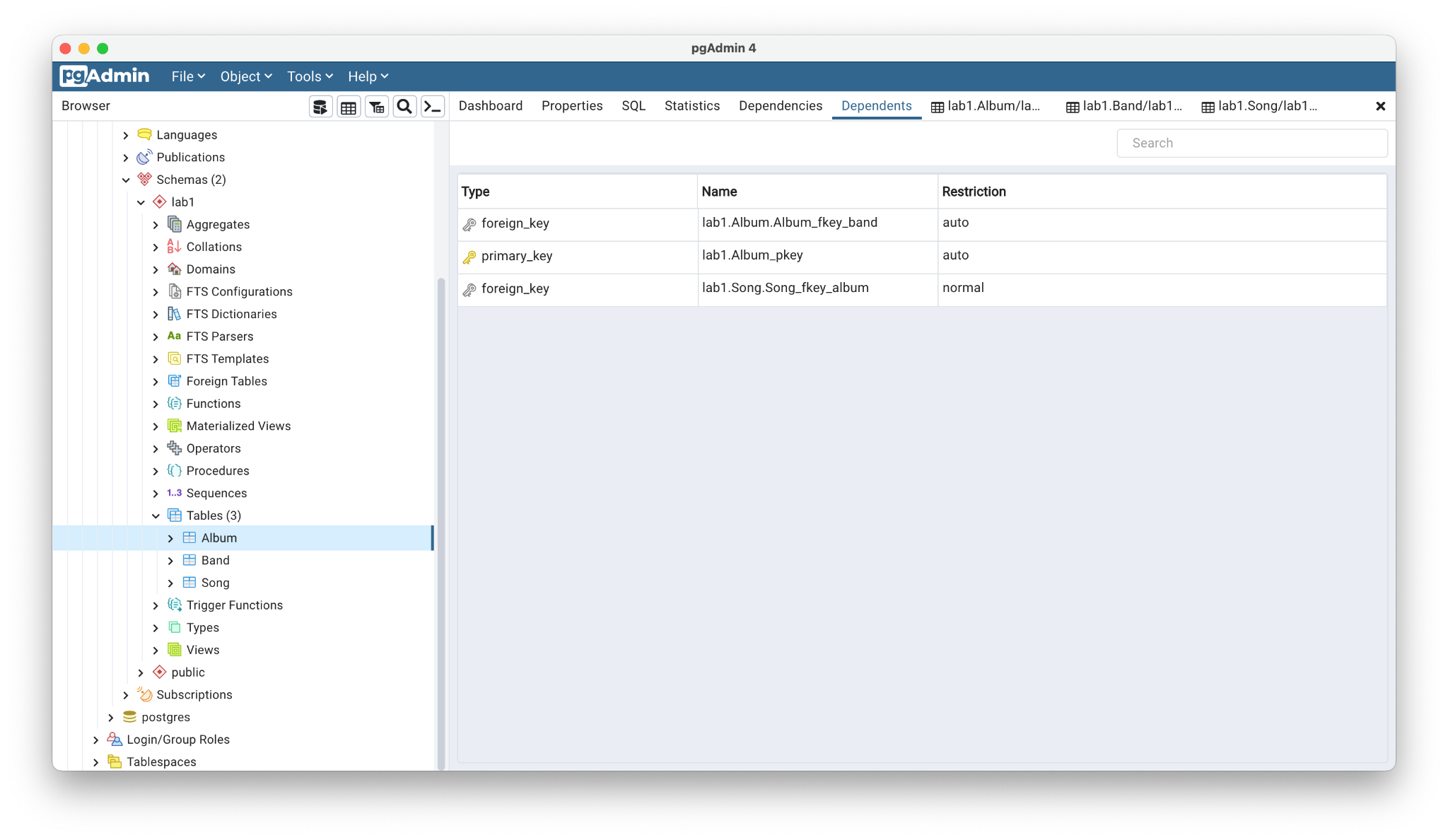
****

**Колонки таблиці “band\_rec”:**

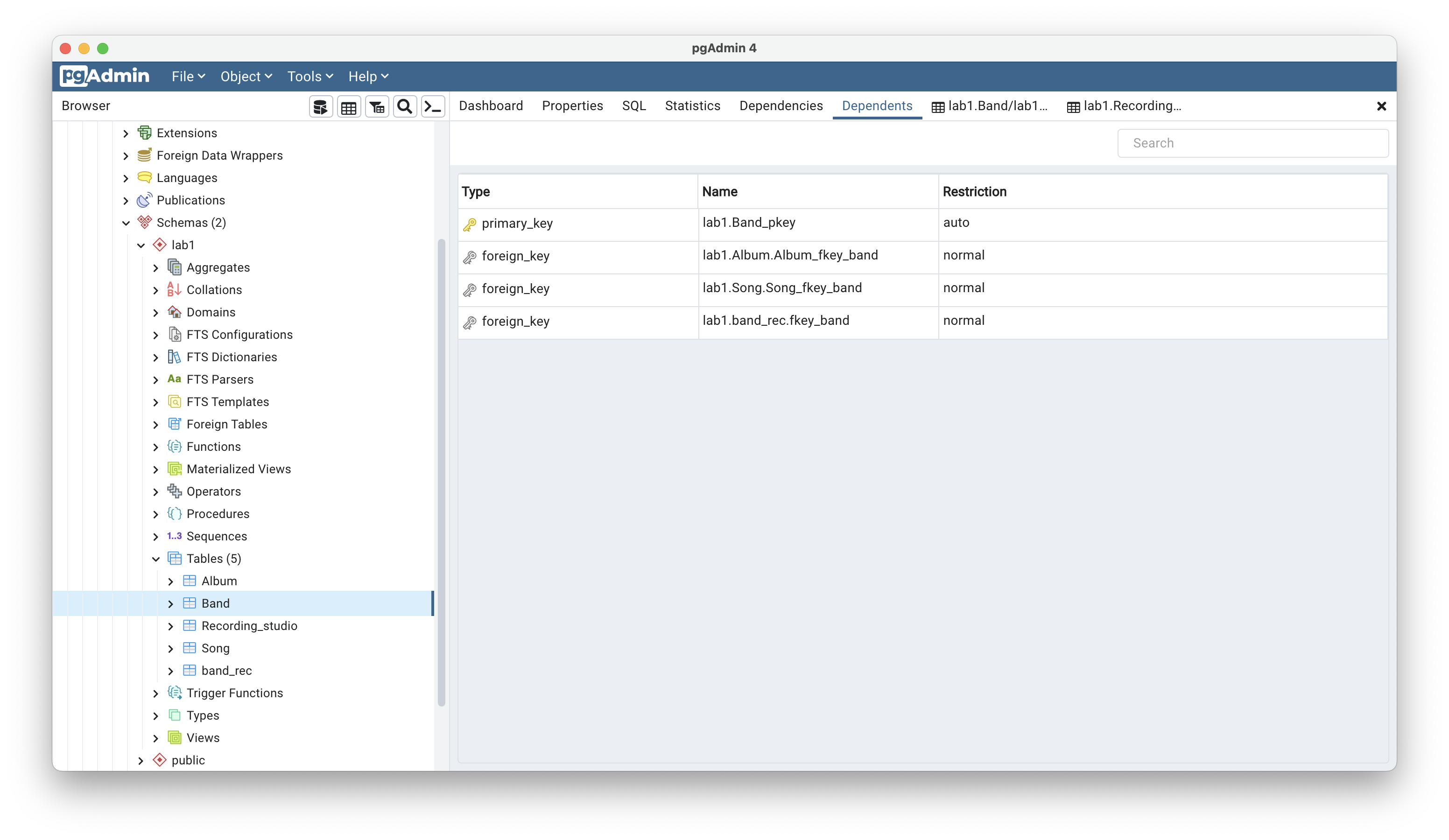


**Список залежностей для таблиці “Album”:**

Первинний ключ: *lab1.Album\_pkey*

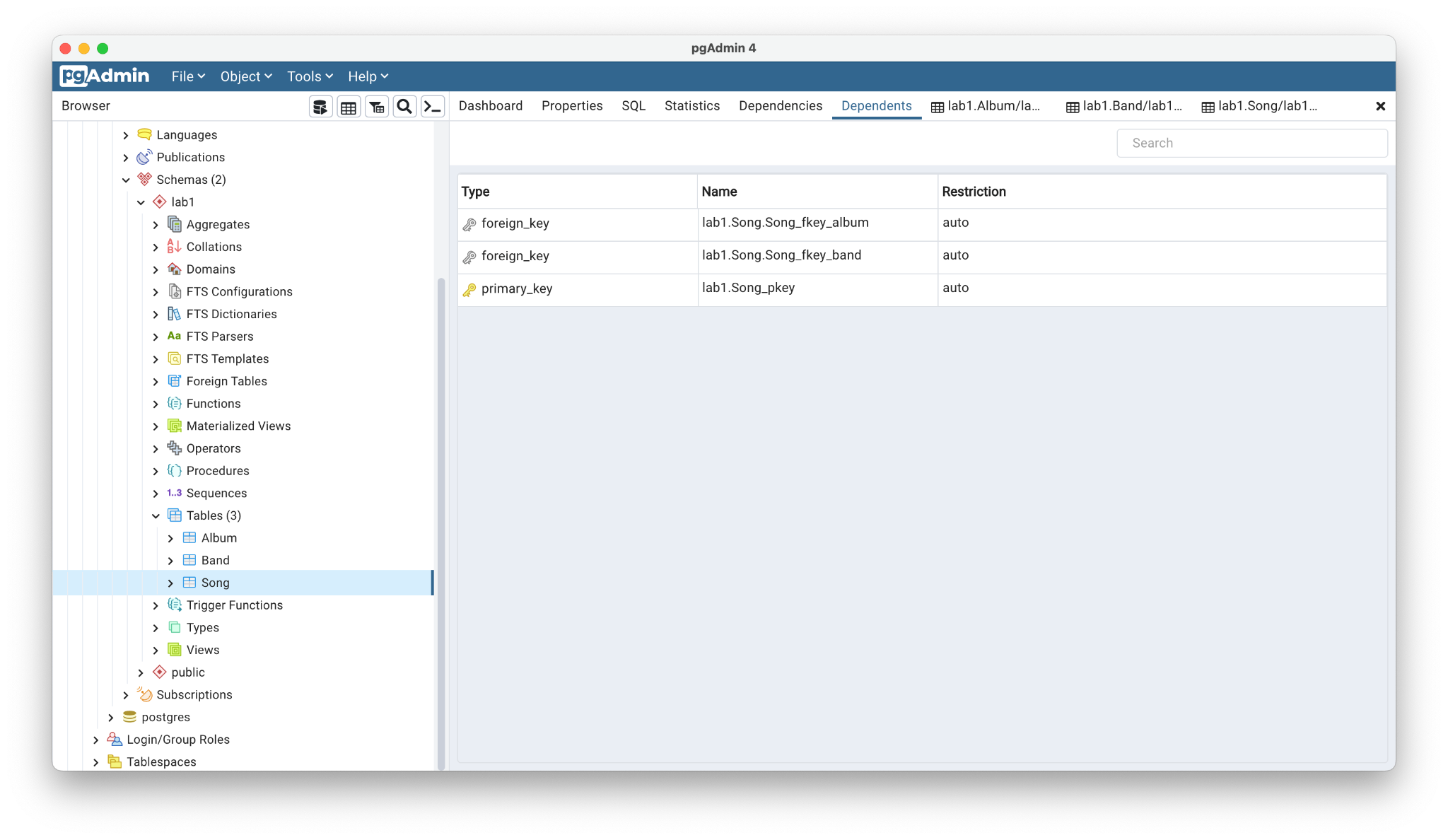
Зовнішній ключ: *lab1.Album.Album\_fkey\_band*

**Список залежностей для таблиці “Band”:**

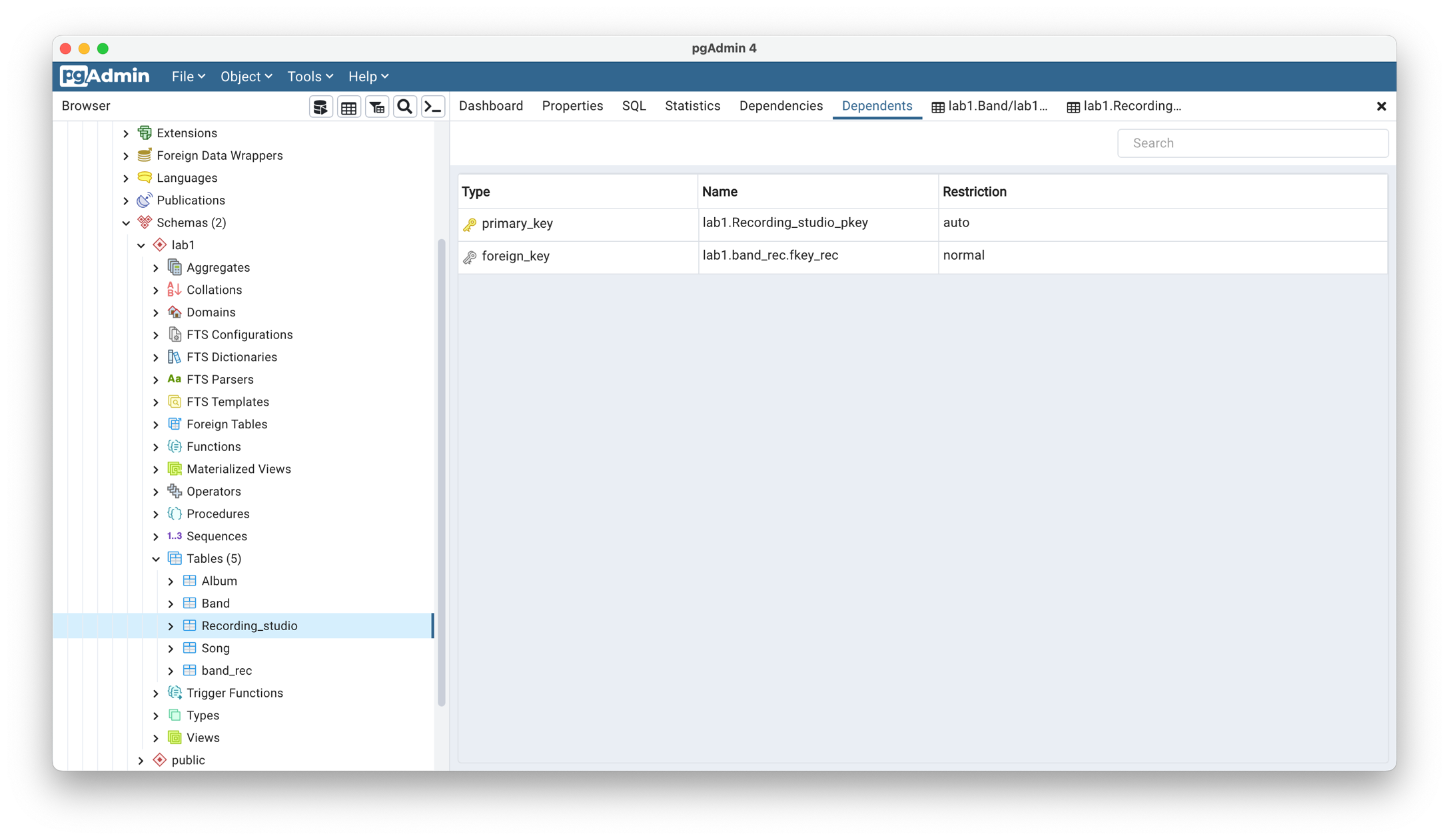
Первинний ключ: *lab1.Band\_pkey*

**Список залежностей для таблиці “Song”:**

Первинний ключ: *lab1.Song\_pkey*

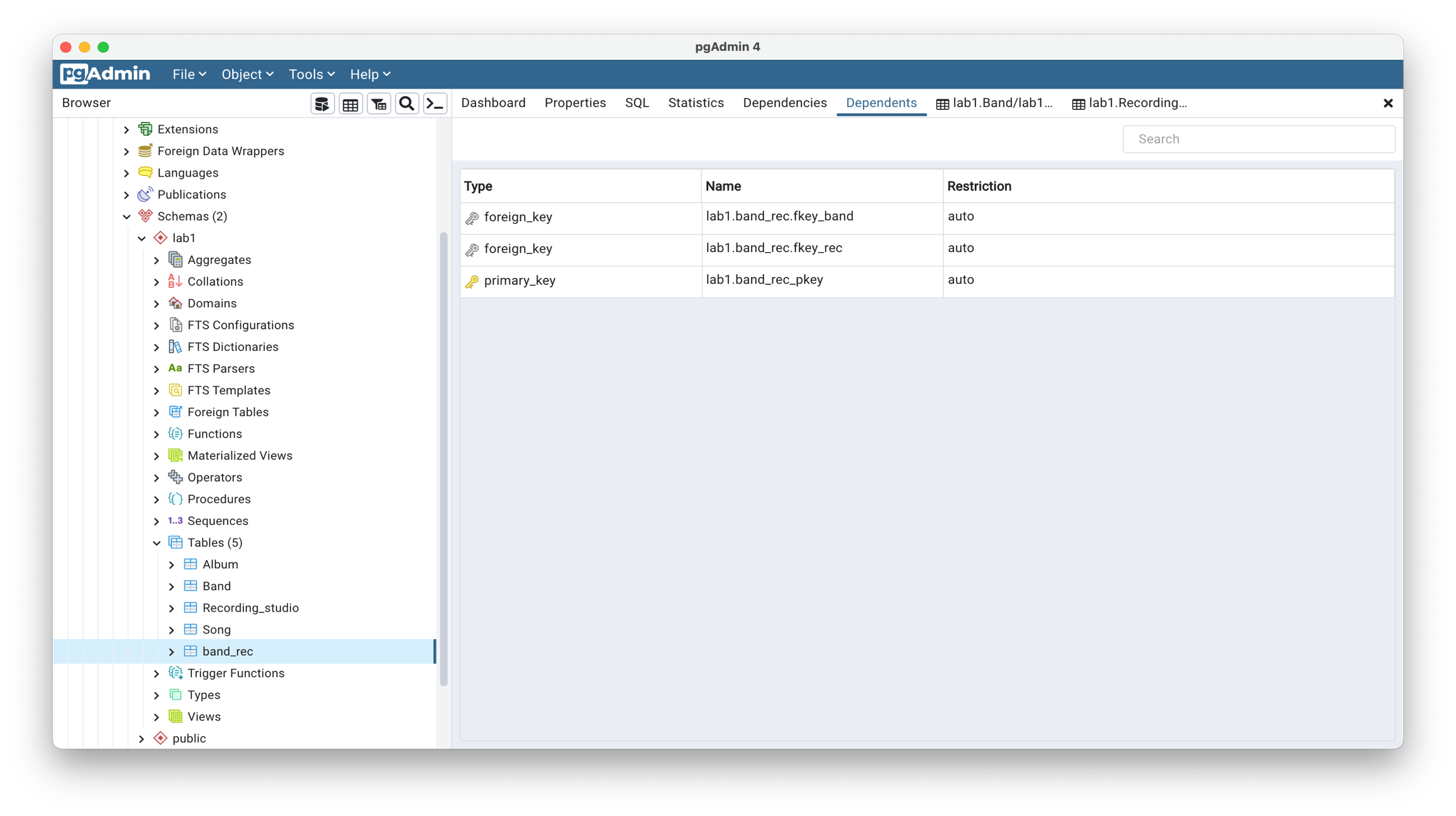
Зовнішній ключ: *lab1.Song.Song\_fkey\_band, lab1.Song.Song\_fkey\_album*

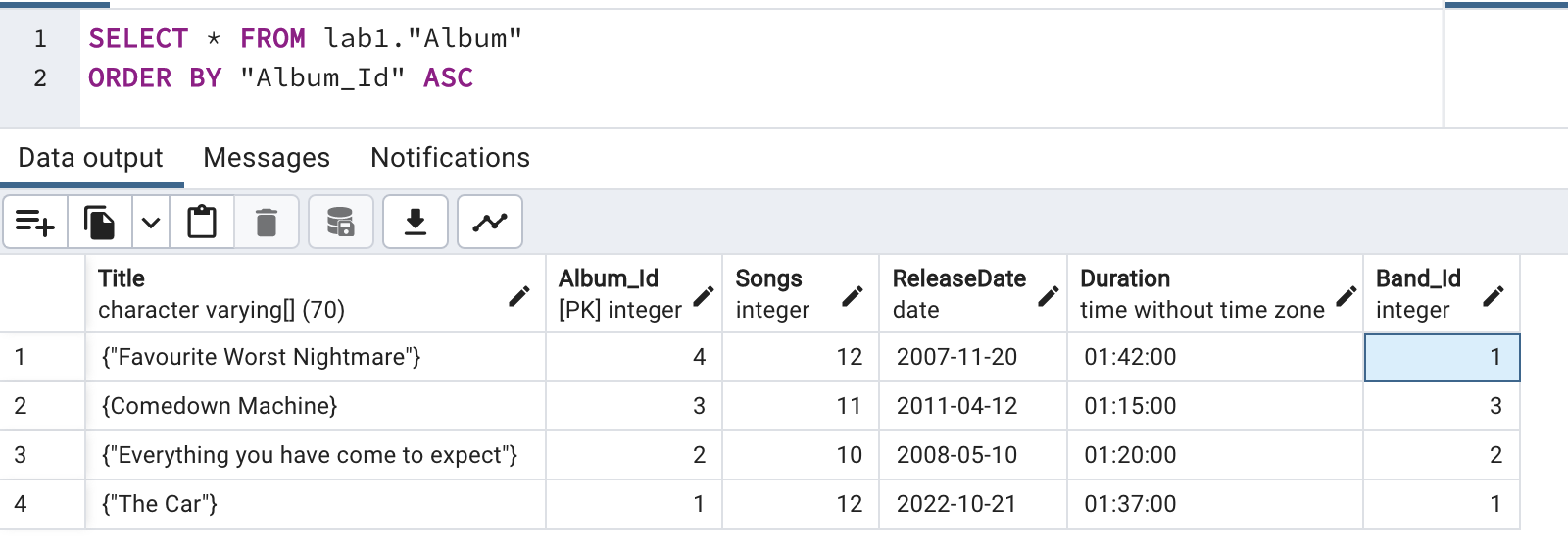
**Список залежностей для таблиці “Recording\_studio”:**

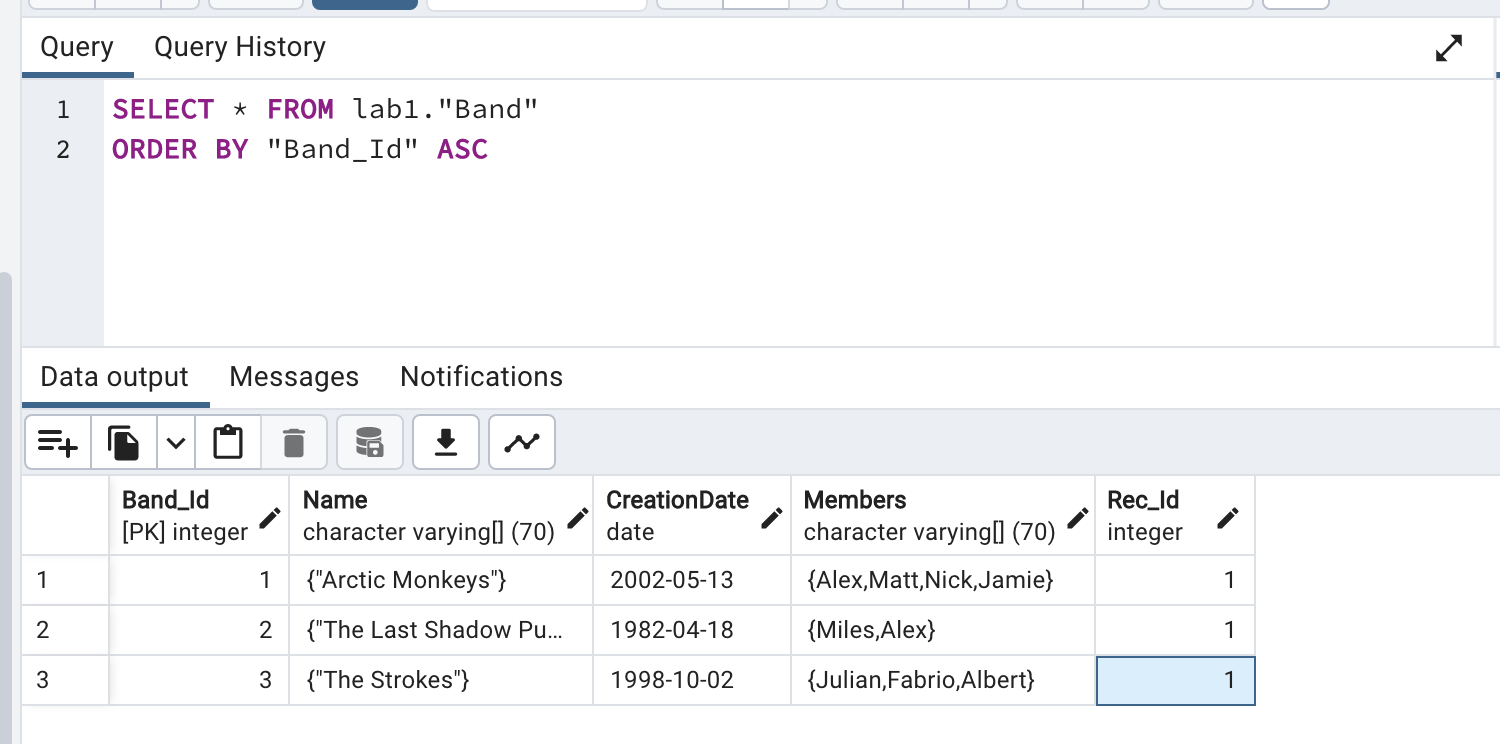
****Первинний ключ: *lab1.Recording\_studio\_pkey*

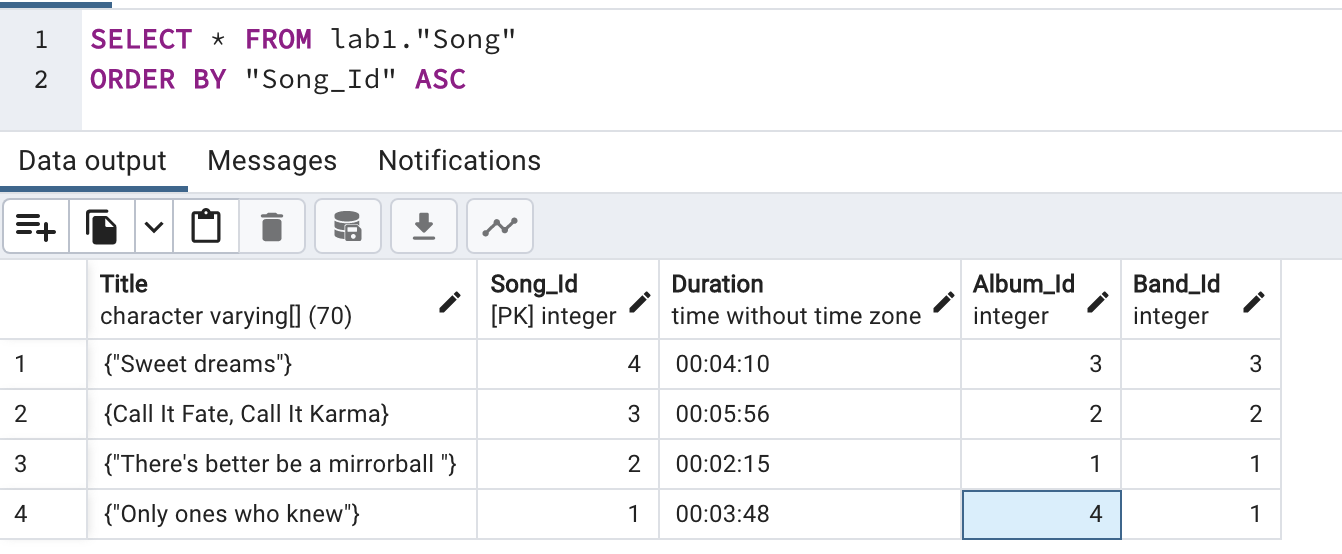
**Список залежностей для таблиці “band\_rec”:**

Первинний ключ: *lab1.band\_rec\_pkey*

****Зовнішній ключ: *lab1.band\_rec.fkey\_band, lab1.band\_rec.fkey\_rec*

**Зміст таблиці “Album”:**

**Зміст таблиці “Band”:**

**Зміст таблиці “Song”:**

**Зміст таблиці “Recording studio”:**



**SQL-текст опису БД «Музичний гурт»**

CREATE TABLE IF NOT EXISTS lab1."Album"

(

"Title" character varying(70)[] COLLATE pg\_catalog."default",

"Album\_Id" integer NOT NULL,

"Songs" integer,

"ReleaseDate" date,

"Duration" time without time zone,

"Band\_Id" integer,

CONSTRAINT "Album\_pkey" PRIMARY KEY ("Album\_Id"),

CONSTRAINT "Album\_fkey\_band" FOREIGN KEY ("Band\_Id")

REFERENCES lab1."Band" ("Band\_Id") MATCH FULL

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS lab1."Album"

OWNER to postgres;

CREATE TABLE IF NOT EXISTS lab1."Band"

(

"Band\_Id" integer NOT NULL,

"Name" character varying(70)[] COLLATE pg\_catalog."default",

"CreationDate" date,

"Members" character varying(70)[] COLLATE pg\_catalog."default",

"Rec\_Id" integer,

CONSTRAINT "Band\_pkey" PRIMARY KEY ("Band\_Id")

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS lab1."Band"

OWNER to postgres;

CREATE TABLE IF NOT EXISTS lab1."Recording\_studio"

(

"Name" character varying(70)[] COLLATE pg\_catalog."default",

"Rec\_Id" integer NOT NULL,

"Band\_Id" integer,

CONSTRAINT "Recording\_studio\_pkey" PRIMARY KEY ("Rec\_Id")

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS lab1."Recording\_studio"

OWNER to postgres;

CREATE TABLE IF NOT EXISTS lab1."Song"

(

"Title" character varying(70)[] COLLATE pg\_catalog."default",

"Song\_Id" integer NOT NULL,

"Duration" time without time zone,

"Album\_Id" integer,

"Band\_Id" integer,

CONSTRAINT "Song\_pkey" PRIMARY KEY ("Song\_Id"),

CONSTRAINT "Song\_fkey\_album" FOREIGN KEY ("Album\_Id")

REFERENCES lab1."Album" ("Album\_Id") MATCH FULL

CONSTRAINT "Song\_fkey\_band" FOREIGN KEY ("Band\_Id")

REFERENCES lab1."Band" ("Band\_Id") MATCH FULL

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS lab1."Song"

OWNER to postgres;

CREATE TABLE IF NOT EXISTS lab1.band\_rec

(

"Band\_Id" integer NOT NULL,

"Rec\_Id" integer NOT NULL,

CONSTRAINT band\_rec\_pkey PRIMARY KEY ("Band\_Id", "Rec\_Id"),

CONSTRAINT fkey\_band FOREIGN KEY ("Band\_Id")

REFERENCES lab1."Band" ("Band\_Id") MATCH SIMPLE

CONSTRAINT fkey\_rec FOREIGN KEY ("Rec\_Id")

REFERENCES lab1."Recording\_studio" ("Rec\_Id") MATCH SIMPLE

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS lab1.band\_rec

OWNER to postgres;