

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:

ФПМ групи KB-02

Дахал К.

Київ 2022

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.

Сутності предметної області

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

1. Гурт (Band), з атрибутами: код гурту, ім'я, учасники, дата створення.

Описує базові дані та склад певного гурту. Кожен гурт має альбоми та пісні.

2. Альбом (Album), з атрибутами: код альбому, 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, назва, тривалість.

Опис певної пісні. Кожна пісня має одного виконавця та походить з одного альбому.

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,

Концептуальна модель предметної області “Музична група”

Концептуальна модель наведена на рисунку 1.

Особливістю діаграми є трьохсторонній зв'язок між сутностями “Гурт”, “Альбом”, “Пісня”.

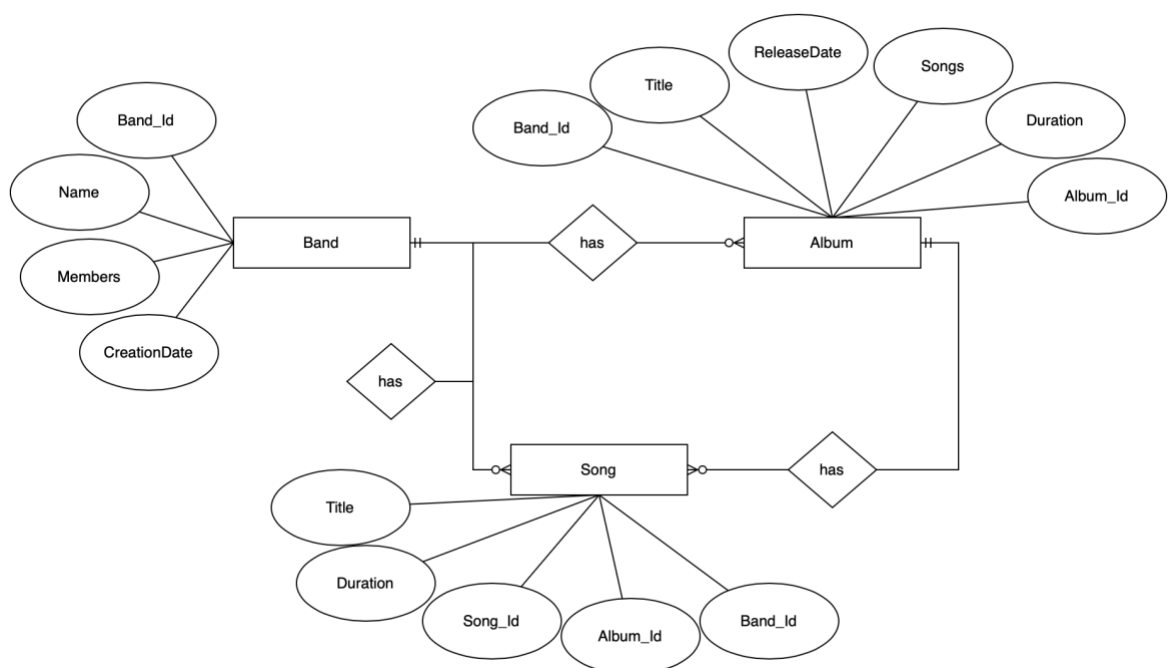


Рис. 1– ER-діаграма за нотацією “Crow’s foot”.

Перетворення концептуальної моделі в логічну схему бази даних

Сутність “Гурт” перетворено у таблицю “Band”.

Сутність “Альбом” перетворено у таблицю “Album”. Band_Id зберігає зовнішній ключ для зв’язку з таблицею Band.

Сутність “Пісня” перетворено у таблицю “Song”. Band_Id зберігає зовнішній ключ для зв’язку з таблицею Band. Album_Id зберігає зовнішній ключ для зв’язку з таблицею Album.

Логічна модель (схема) БД «Кінотеатр»

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

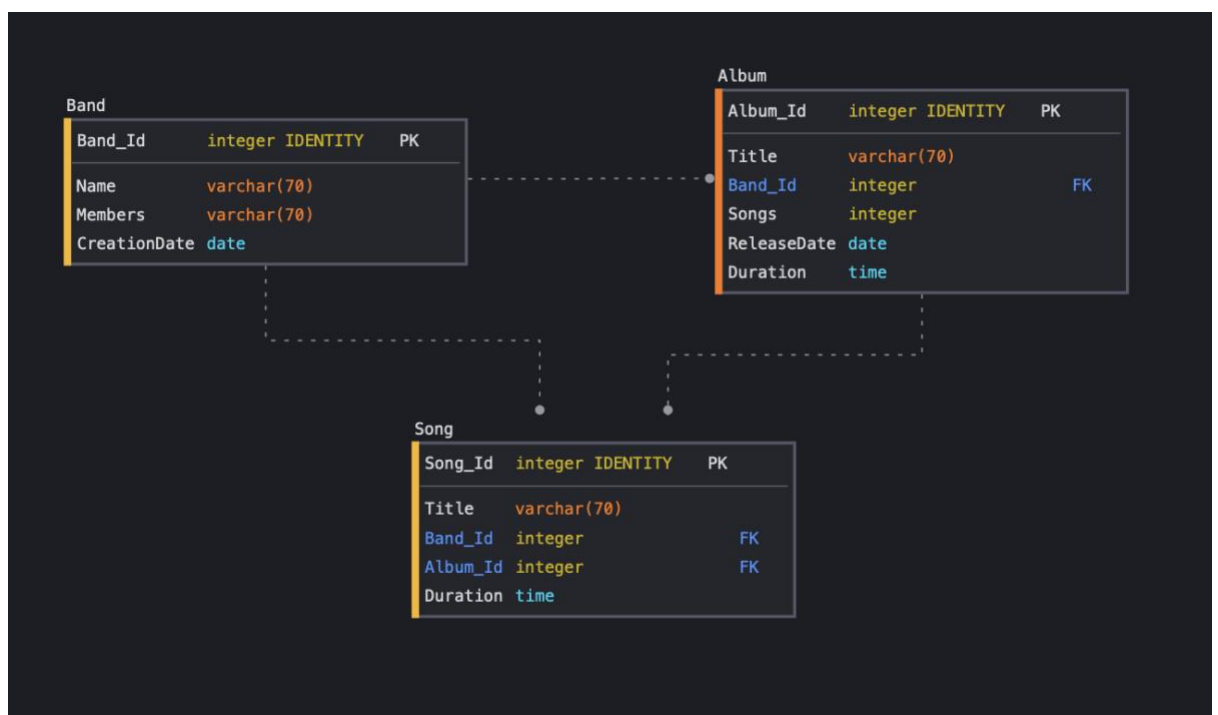


Рис. 2 – Схема бази даних (інструмент: sqldbm.com)

Опис об'єктів бази даних у вигляді таблиці

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)

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 (дата 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)

Відповідність схеми бази даних до третьої нормальної форми

Схема відповідає 1НФ, тому що в таблиці немає дубльованих рядків, в кожному стовпці зберігаються дані одного типу і в кожній комірці зберігається скалярне значення.

Схема відповідає 2НФ, тому що вона має первинний ключ, а всі не ключові стовпці таблиці залежать від первинного ключа.

Схема відповідає 3НФ, тому що кожен неключовий атрибут безпосередньо залежить від первинного ключа.

Фізична модель БД «Музичний гурт» у pgAdmin4

Колонки таблиці “Album”:

Album

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Inherited from table(s)

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
		Title	character varying[]	70	<input type="checkbox"/>	<input type="checkbox"/>	
		Album_Id	integer		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		Songs	integer		<input type="checkbox"/>	<input type="checkbox"/>	
		ReleaseDate	date		<input type="checkbox"/>	<input type="checkbox"/>	
		Duration	time without time zone		<input type="checkbox"/>	<input type="checkbox"/>	
		Band_Id	integer		<input type="checkbox"/>	<input type="checkbox"/>	

Колонки таблиці “Band”:

Band

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Inherited from table(s)

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
		Band_Id	integer		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		Name	character varying[]	70	<input type="checkbox"/>	<input type="checkbox"/>	
		CreationDate	date		<input type="checkbox"/>	<input type="checkbox"/>	
		Members	character varying[]	70	<input type="checkbox"/>	<input type="checkbox"/>	

Колонки таблиці “Song”:

Song

GeneralColumnsAdvancedConstraintsParametersSecuritySQL

Inherited from table(s) Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	Title	character varying[]	70		<input type="checkbox"/>	<input type="checkbox"/>	
	Song_Id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	Duration	time without time zone			<input type="checkbox"/>	<input type="checkbox"/>	
	Album_Id	integer			<input type="checkbox"/>	<input type="checkbox"/>	
	Band_Id	integer			<input type="checkbox"/>	<input type="checkbox"/>	

Первинний ключ для таблиці “Album”: lab1.Album_pkey
Зовнішній ключ для таблиці “Album”: lab1.Album.Album_fkey_band

pgAdmin 4

pgAdminFileObjectToolsHelp

BrowserDashboardPropertiesSQLStatisticsDependencieslab1.Album/la...lab1.Band/lab1...lab1.Song/lab1...

Search



Type	Name	Restriction
foreign_key	lab1.Album.Album_fkey_band	auto
primary_key	lab1.Album_pkey	auto
foreign_key	lab1.Song.Song_fkey_album	normal

Album

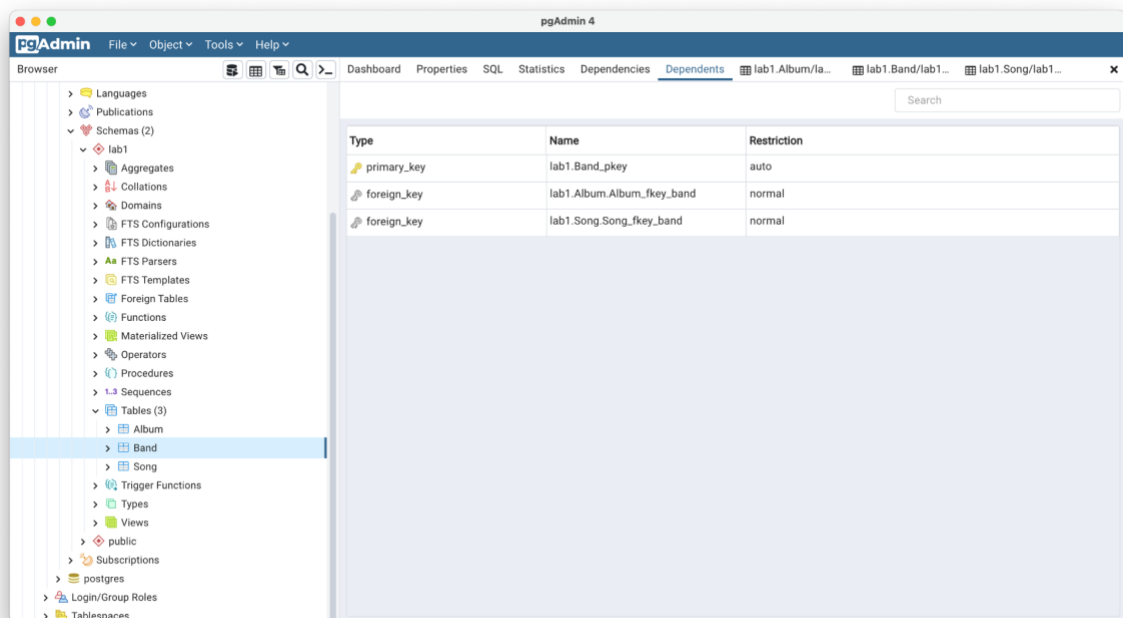
GeneralColumnsAdvancedConstraintsParametersSecuritySQL



Primary KeyForeign KeyCheckUniqueExclude

	Name	Columns
	Album_pkey	Album_Id

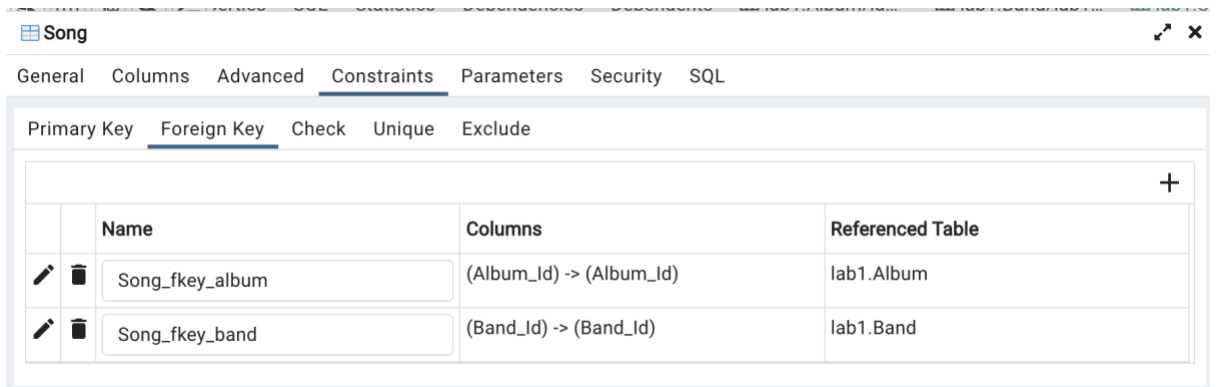
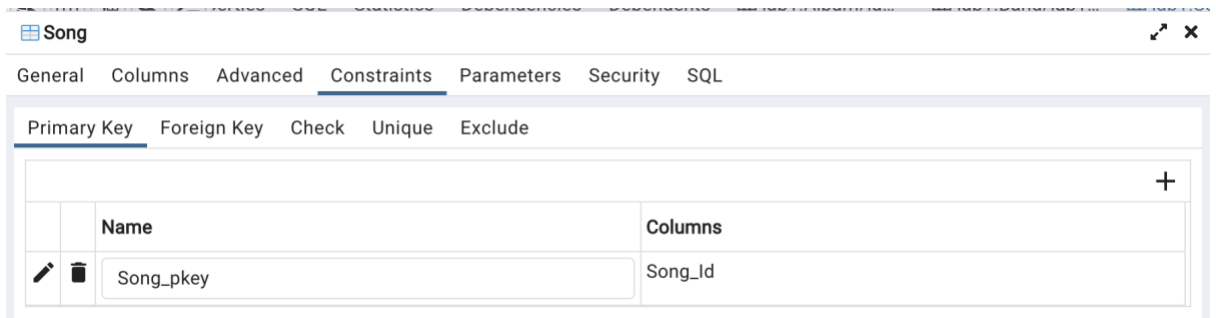
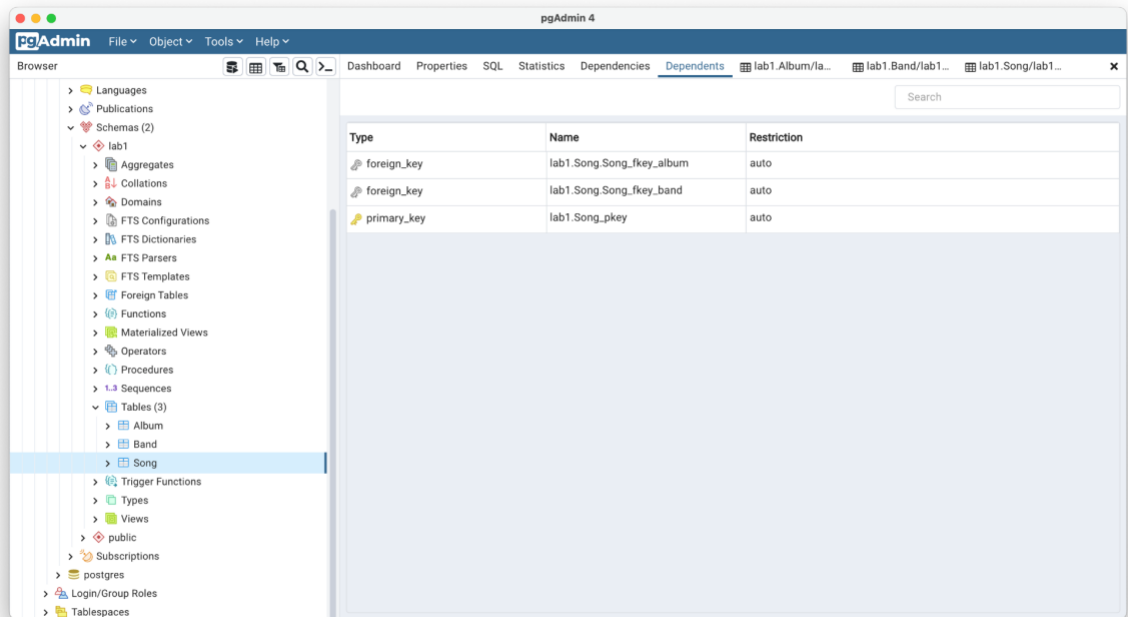
Album			
<div>General Columns Advanced <u>Constraints</u> Parameters Security SQL</div>			
<div>Primary Key <u>Foreign Key</u> Check Unique Exclude</div>			
+			
	Name	Columns	Referenced Table
 	Album_fkey_band	(Band_Id) -> (Band_Id)	lab1.Band

Первинний ключ для таблиці “Band”: **lab1.Band_pkey**



Band		
<div>General Columns Advanced <u>Constraints</u> Parameters Security SQL</div>		
<div>Primary Key <u>Foreign Key</u> Check Unique Exclude</div>		
+		
	Name	Columns
 	Band_pkey	Band_Id

Первинний ключ для таблиці “Song”: **lab1.Song_pkey**
Зовнішній ключ для таблиці “Song”: **lab1.Song.Song_fkey_band,**
lab1.Song.Song_fkey_album



Зміст таблиці “Album”:

```
1 SELECT * FROM lab1."Album"  
2 ORDER BY "Album_Id" ASC
```

Data output Messages Notifications

	Title character varying[] (70)	Album_Id [PK] integer	Songs integer	ReleaseDate date	Duration time without time zone	Band_Id integer
1	{"Favourite Worst Nightmare"}	4	12	2007-11-20	01:42:00	1
2	{Comedown Machine}	3	11	2011-04-12	01:15:00	3
3	{"Everything you have come to expect"}	2	10	2008-05-10	01:20:00	2
4	{"The Car"}	1	12	2022-10-21	01:37:00	1

Зміст таблиці “Band”:

```
1 SELECT * FROM lab1."Band"  
2 ORDER BY "Band_Id" ASC
```

Data output Messages Notifications

	Band_Id [PK] integer	Name character varying[] (70)	CreationDate date	Members character varying[] (70)
1	3	{"The Strokes"}	1998-10-02	{Julian,Fabrio,Albert}
2	2	{"The Last Shadow Puppets"}	1982-04-18	{Miles, Alex}
3	1	{"Arctic Monkeys"}	2002-05-13	{Alex,Matt,Nick,Jamie}

Зміст таблиці “Song”:

```
1 SELECT * FROM lab1."Song"  
2 ORDER BY "Song_Id" ASC
```

Data output Messages Notifications

	Title character varying[] (70)	Song_Id [PK] integer	Duration time without time zone	Album_Id integer	Band_Id integer
1	{"Sweet dreams"}	4	00:04:10	3	3
2	{Call It Fate, Call It Karma}	3	00:05:56	2	2
3	{"There's better be a mirrorball "}	2	00:02:15	1	1
4	{"Only ones who knew"}	1	00:03:48	4	1

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",
    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."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;
```