**Лабораторна робота №1**

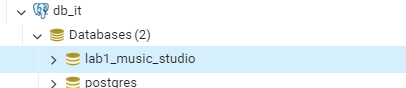
**Звіт**

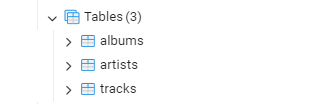
З дисципліни “Бази даних та інформаційні системи”

МІТ-31, Бессараб Антон

**Варіант 6**: Музична студія (таблиці: альбоми, виконавці, треки).

Перш за все ми створюємо локальний сервер , після цього в ньому створюємо базу данних для першої лабораторної роботи.



У цій базі данних ми створюємо три таблиці згідно нашого варіанту - albums, artists, tracks  


Створення бази данних і табличок:

CREATE DATABASE lab1\_music\_studio

WITH

OWNER = postgres

ENCODING = 'UTF8'

LOCALE\_PROVIDER = 'libc'

CONNECTION LIMIT = -1

IS\_TEMPLATE = False;

CREATE TABLE public.artists

(

artist\_id serial,

name character varying(150) NOT NULL,

country character varying(150),

age integer,

PRIMARY KEY (artist\_id)

);

ALTER TABLE IF EXISTS public.artists

OWNER to postgres;

CREATE TABLE public.albums

(

album\_id serial,

tittle character varying(150) NOT NULL,

artist\_id integer,

release\_date date,

genre character varying(150),

CONSTRAINT album\_id PRIMARY KEY (album\_id),

CONSTRAINT artist\_id FOREIGN KEY (artist\_id)

REFERENCES public.artists (artist\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE

NOT VALID

);

ALTER TABLE IF EXISTS public.albums

OWNER to postgres;

CREATE TABLE public.tracks

(

track\_id serial,

tittle character varying(150) NOT NULL,

album\_id integer,

duration interval,

CONSTRAINT track\_id PRIMARY KEY (track\_id),

CONSTRAINT album\_id FOREIGN KEY (album\_id)

REFERENCES public.albums (album\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE

NOT VALID

);

ALTER TABLE IF EXISTS public.tracks

OWNER to postgres;

Зв’язки :

1. Один виконавець (artists.id) може мати багато альбомів (albums.artist\_id).
2. Один альбом (albums.id) може містити багато треків (tracks.album\_id).

Далі за допомогою запитів INSERT ми наповнюємо таблиці данними:

INSERT INTO

ARTISTS (NAME, COUNTRY, AGE)

VALUES

('Coldplay', 'United Kingdom', 27),

('Eminem', 'United States', 51),

('The Weeknd', 'Canada', 33);

INSERT INTO

ALBUMS (TITTLE, ARTIST\_ID, RELEASE\_DATE, GENRE)

VALUES

('Parachutes', 10, '2000-07-10', 'Alternative Rock'),

('The Eminem Show', 11, '2002-05-26', 'Hip-Hop'),

('After Hours', 12, '2020-03-20', 'R&B');

INSERT INTO

TRACKS (TITTLE, ALBUM\_ID, DURATION)

VALUES

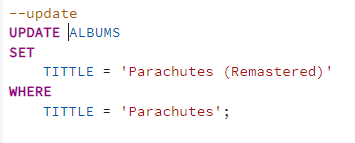
('Yellow', 13, '00:04:29'),

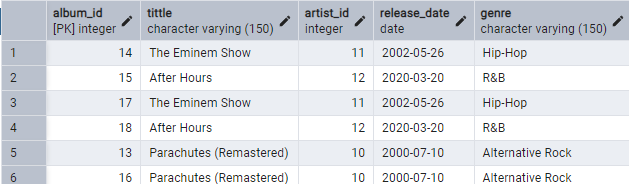
('Without Me', 14, '00:04:50'),

('Blinding Lights', 15, '00:03:20');

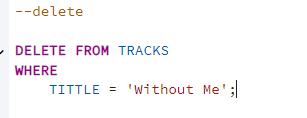
За допомогою запитів SELECT перевіряємо наповнення таблиць.

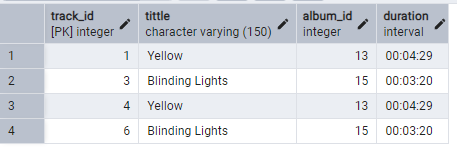
Далі за допомогою запитів UPDATE змінюємо назву альбому з Parachutes на Parachutes (Remastered





Використовуючи DELETE видаляємо всі треки з назвою Without me





Висновок: у ході виконання лабораторної роботи було розглянуто основи роботи з PostgreSQL. Було здійснено встановлення та налаштування СУБД, що є першим кроком для роботи з реляційними базами даних.

Було створено базу даних "lab1\_music\_studio" та необхідні таблиці (artists, albums, tracks) із використанням первинних та зовнішніх ключів для забезпечення зв’язків між сутностями. Окрім створення таблиць, було виконано наповнення бази даних за допомогою INSERT, а також оновлення та видалення записів через UPDATE і DELETE.