



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**Кафедра системного програмування та спеціалізованих комп'ютерних
систем**

Лабораторна робота №2

з дисципліни **Бази даних і засоби управління**

на тему: *“Створення додатку бази даних, орієнтованого на взаємодію з СУБД
PostgreSQL”*

Виконав:
студент III курсу
групи KB-92
Орлов Є. А.
Перевірив:
Петрашенко А. В.

Київ – 2021

Постановка задачі

Метою роботи є здобуття вмінь програмування прикладних додатків баз даних PostgreSQL.

Загальне завдання роботи полягає у наступному:

1. Реалізувати функції перегляду, внесення, редагування та вилучення даних у таблицях бази даних, створених у лабораторній роботі №1, засобами консольного інтерфейсу.
2. Передбачити автоматичне пакетне генерування «рандомізованих» даних у базі.
3. Забезпечити реалізацію пошуку за декількома атрибутами з двох та більше сутностей одночасно: для числових атрибутів – у рамках діапазону, для рядкових – як шаблон функції LIKE оператора SELECT SQL, для логічного типу – значення True/False, для дат – у рамках діапазону дат.
4. Програмний код виконати згідно шаблону MVC (модель-подання-контролер).

Інформація про програму

Посилання на репозиторій у GitHub з вихідним кодом програми та звітом:

<https://github.com/CoachJohnsy/PostgreSQL-basics/tree/master/lab-2>

Використана мова програмування:

Python 3.10

Використані бібліотеки:

psycopg2 (для зв'язку з СУБД),

numpy, collections (для більш зручних операцій маніпулювання даними в контейнерах)

xml (для наповнення бази даних реальними даними з xml документу iTunes бібліотеки пісень)

re (для використання регулярних виразів під час валідації та обмеження вводу даних форматно)

pandas, tabulate (для створення потрібного формату виводу даних до користувача)

typing (для типізації змінних та функцій)

time (для виміру часу запиту пошуку для завдання 3)

Модель «сутність-зв'язок» галузі музичної бібліотеки

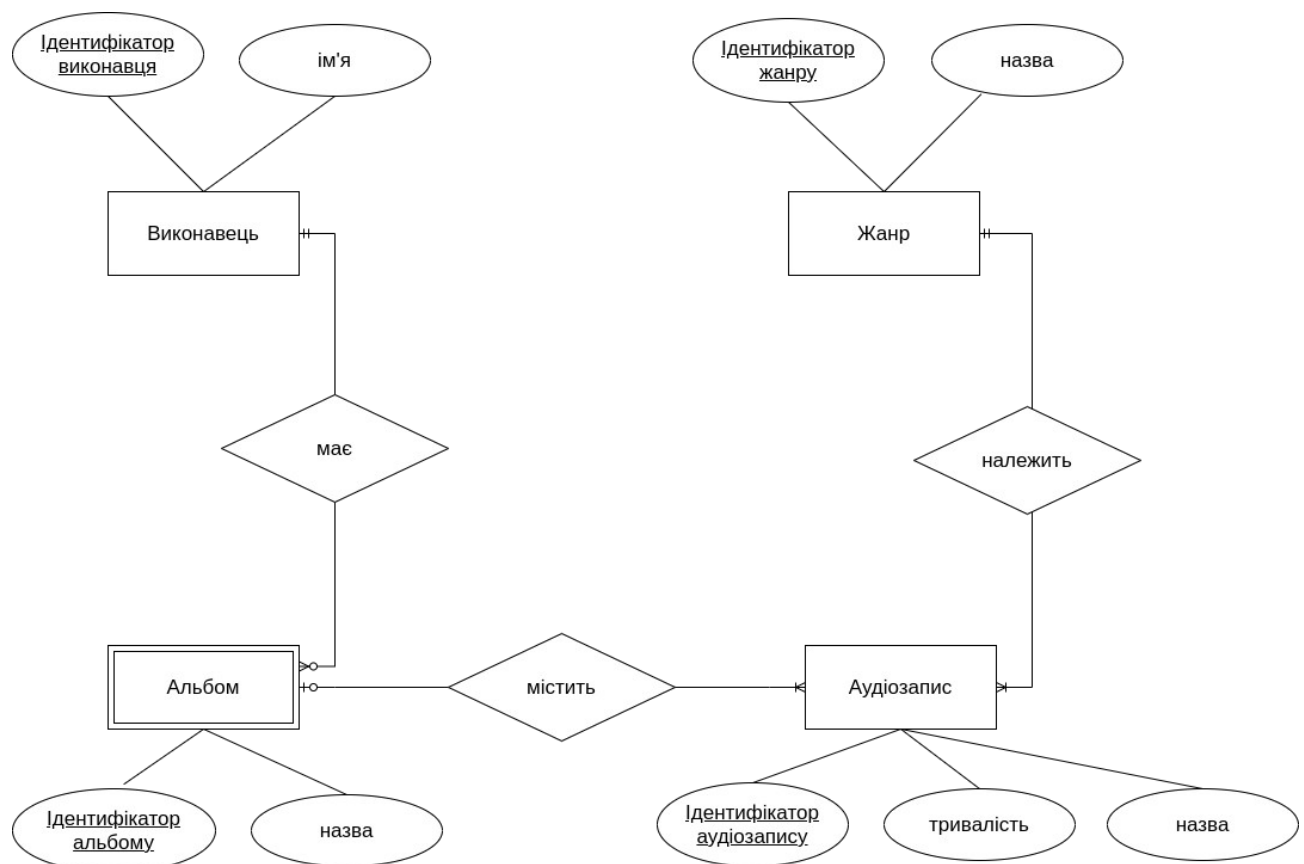


Рисунок 1. ER-діаграма, побудована за нотацією Чена

Перелік сутностей з їх призначенням

1. Виконавець, з атрибутами: ідентифікатор, ім'я. Призначена для збереження переліку артистів в бібліотеці користувача.
2. Альбом, з атрибутами: ідентифікатор, назва. Призначена для збереження списку назв музичних альбомів в бібліотеці користувача.
3. Жанр, з атрибутами: ідентифікатор, назва. Призначена для збереження списку назв жанрів пісень, які є в бібліотеці користувача.
4. Аудіозапис, з атрибутами: ідентифікатор, назва, довжина. Призначена для збереження списку пісень та аудіозаписів, які є в бібліотеці користувача.

Перетворення моделі у схему бази даних

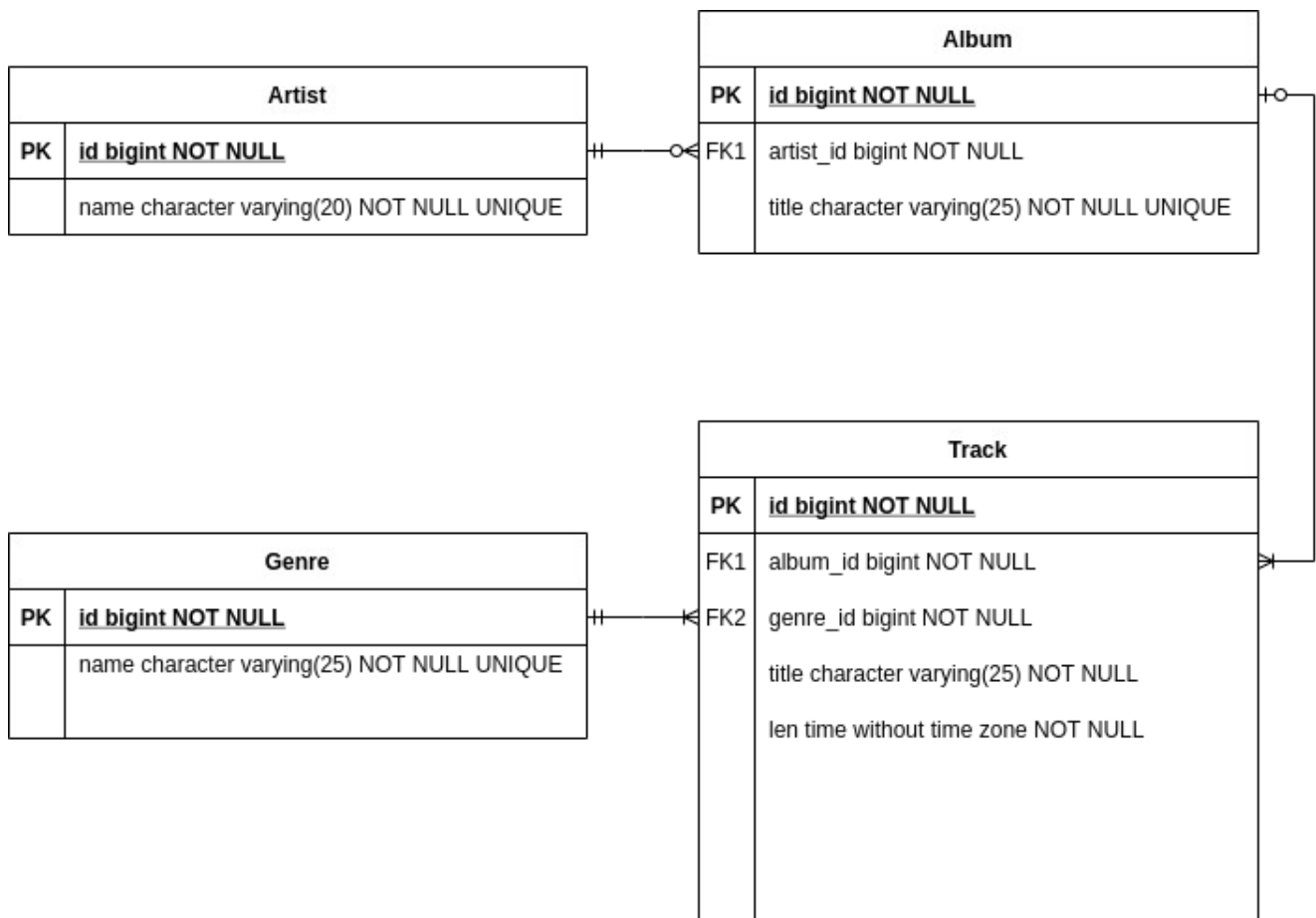


Рисунок 2. Схема бази даних

Характеристика атрибутів (назви, типи, обмеження)

music_library=# \d genre

Table "public.genre"				
Column	Type	Collation	Nullable	Default
id	bigint		not null	nextval('genre_id_seq'::regclass)
name	character varying(25)		not null	

Indexes:

"genre_pkey" PRIMARY KEY, btree (id)

"genre_name_key" UNIQUE CONSTRAINT, btree (name)

Referenced by:

TABLE "track" CONSTRAINT "track_genre_id_fkey" FOREIGN KEY (genre_id) REFERENCES genre(id)

music_library=#

music_library=# \d artist

Table "public.artist"				
Column	Type	Collation	Nullable	Default
id	bigint		not null	nextval('artist_id_seq'::regclass)
name	character varying(20)		not null	

Indexes:

"artist_pkey" PRIMARY KEY, btree (id)

"unique_artist_name" UNIQUE CONSTRAINT, btree (name)

Referenced by:

TABLE "album" CONSTRAINT "album_artist_id_fkey" FOREIGN KEY (artist_id) REFERENCES artist(id)

music_library=#

music_library=# \d album

Table "public.album"				
Column	Type	Collation	Nullable	Default
id	bigint		not null	nextval('album_id_seq'::regclass)
title	character varying(25)		not null	
artist_id	bigint		not null	

Indexes:

"album_pkey" PRIMARY KEY, btree (id)

"album_title_key" UNIQUE CONSTRAINT, btree (title)

Foreign-key constraints:

"album_artist_id_fkey" FOREIGN KEY (artist_id) REFERENCES artist(id)

Referenced by:

TABLE "track" CONSTRAINT "track_album_id_fkey" FOREIGN KEY (album_id) REFERENCES album(id)

music_library=#

music_library=# \d track

Table "public.track"				
Column	Type	Collation	Nullable	Default
id	bigint		not null	nextval('track_id_seq'::regclass)
title	character varying(25)		not null	
len	time without time zone		not null	
album_id	bigint		not null	
genre_id	bigint		not null	

Indexes:

"track_pkey" PRIMARY KEY, btree (id)

Foreign-key constraints:

"track_album_id_fkey" FOREIGN KEY (album_id) REFERENCES album(id)

"track_genre_id_fkey" FOREIGN KEY (genre_id) REFERENCES genre(id)

music_library=#

Приклад вмісту таблиць бази даних

```
music_library=# TABLE artist;
```

id	name
----	------

1	AC/DC
---	-------

2	Metallica
---	-----------

3	Nina Simone
---	-------------

4	Ricky Martin
---	--------------

(4 rows)

```
music_library=#
```

```
music_library=#
```

```
music_library=#
```

```
music_library=#
```

```
music_library=# TABLE genre;
```

id	name
----	------

1	Rock
---	------

2	Metal
---	-------

3	Jazz
---	------

4	Pop
---	-----

(4 rows)

```
music_library=#
```

```
music_library=#
```

```
music_library=#
```

```
music_library=#
```

```
music_library=# TABLE album;
```

id	title	artist_id
----	-------	-----------

1	Black in Black	1
---	----------------	---

2	Life	4
---	------	---

3	Power Up	1
---	----------	---

4	Baltimore	3
---	-----------	---

5	...And Justice For All	2
---	------------------------	---

(5 rows)

```
music_library=#
```

```
music_library=#
```

```
music_library=#
```

```
music_library=#
```

```
music_library=# TABLE track;
```

id	title	len	album_id	genre_id
----	-------	-----	----------	----------

1	One	00:07:26	5	2
---	-----	----------	---	---

2	Til I Getto You	00:04:56	2	4
---	-----------------	----------	---	---

3	Shoot to Thrill	00:05:18	1	1
---	-----------------	----------	---	---

4	Baltimore	00:04:38	4	3
---	-----------	----------	---	---

5	Shot in the Dark	00:03:05	3	1
---	------------------	----------	---	---

(5 rows)

Схема меню користувача

Action	Command
INSERT	1
SELECT	2
UPDATE	3
DELETE	4
COMPREHENSIVE	5
GENERATION	6

Enter the action Command: |

1. INSERT – Програма пропонує користувачу додати до бази даних дані використовуючи таблицю з пропованих, вказуючи значення атрибуту id, або без нього.

```
Enter the action Command: 1
Enter table name from available ['artist' 'album' 'track' 'genre']: artist
Enter attribute[s] name from available ('id', 'name'),
to insert data within the entered column[s]. If you want to insert values by several attributes
-> separate attributes by comma ',': id, name
Enter attribute[s] name from available ('id', 'name'),
to insert data within the entered column[s]. If you want to insert values by several attributes
-> separate values by comma ',': 100000, inserted
```

Результат:

```
INSERT 0 1
Execution time: 0.00027751922607421875 seconds
[INFO] Do you want to continue? [Y/n] ☐
```

2. SELECT - Програма пропонує користувачу отримати з бази даних дані використовуючи таблицю з пропованих. Функціонал підтримує отримання усієї таблиці, рядків одного стовпця або декількох.

```

Enter the action Command: 2
Enter table name from available ['artist' 'album' 'track' 'genre']: track
Enter attribute[s] name from available ('id', 'title', 'len', 'year', 'number_within_album', 'album_id', 'genre_id'),
if you want to select particular column[s]. If not -> press 'Enter' otherwise: id, year, number_within_album
+-----+-----+-----+
|  id |  year | number_within_album |
|-----+-----+-----|
|  1 |  1969 |          9 |
|  2 |  1969 |          8 |
|  3 |  1970 |         11 |
|  4 |  1968 |         12 |
|  5 |  1969 |         15 |
|  6 |  1968 |         13 |
|  7 |  1969 |          2 |
|  8 |  1971 |         14 |
|  9 |  1970 |         16 |
| 10 |  1969 |          1 |
| 11 |  1975 |         17 |
| 12 |  1970 |         21 |
| 13 |  1969 |         19 |
| 14 |  1970 |         18 |
| 15 |  1972 |         22 |
| 16 |  1969 |         23 |
| 17 |  1968 |          3 |
| 18 |  1970 |         24 |

```

(Список йде донизу ...)

```

| 3402 | 2012 |          5 |
| 3403 | 2012 |          6 |
| 3404 | 2012 |          7 |
| 3405 | 2012 |          9 |
| 3406 | 2012 |          2 |
| 3414 | 2012 |         10 |
| 3415 | 2012 |         11 |
| 3416 | 2012 |         13 |
| 3417 | 2012 |         14 |
| 3424 | 2021 |          1 |
+-----+-----+-----+
Execution time: 0.0005381107330322266 seconds
[INFO] Do you want to continue? [Y/n]

```


3. UPDATE - Програма пропонує користувачу оновити дані з бази даних використовуючи таблицю з пропаних. Функціонал підтримує оновлення одного або декількох атрибутів, використовуючи одне значення атрибуту таблиці як фільтр.

```
Enter the action Command: 3
Enter table name from available ['artist' 'album' 'track' 'genre']: track
Enter attribute name to update
If you want to update several attributes -> separate them by comma ',': title, len, year
Enter value to assign
If you are updating by several attributes -> separate values by comma ',': updated, 00:59:59, 1001
Enter attribute name as an update filter: len
Enter value as an update filter: 00:00:10
```

```
UPDATE 4
Execution time: 0.0016062259674072266 seconds
[INFO] Do you want to continue? [Y/n] ☐
```

4. DELETE - Програма пропонує користувачу видалити дані з бази даних використовуючи таблицю з пропаних. Функціонал підтримує видалення усіх рядків таблиці або використання значення за атрибутом як фільтр.

```
Enter the action Command: 4
Enter table name from available ['artist' 'album' 'track' 'genre']: track
Enter attribute name from available ('id', 'title', 'len', 'year', 'number_within_album', 'album_id', 'genre_id'),
if you want to delete row[s] by value in column.
If not -> press 'Enter' otherwise: len
Enter value to delete by: 00:00:00
```

```
DELETE 4
Execution time: 0.0008862018585205078 seconds
[INFO] Do you want to continue? [Y/n]
```

5. COMPREHENSIVE – Режим який підтримує більший діапазон використання синтаксису мови sql. Наразі розроблений лише режим SELECT. Він дозволяє отримати дані з декількох таблиць за 3 діапазовими фільтрами.

Рік пісні. Довжина пісні. Кількість пісень в альбомі.

Приклад:

```
Enter the action Command: 5
Choose from available modes ('SELECT',): SELECT
Choose from available options ('year range', 'length range', 'album size'): year range
Enter minimal and maximum publishing years, separated by space

1964 1966
```

Group Name	Album Title	Song Title	Genre	Year
The Rolling Stones	Hot Rocks 1964-1971	(I Can't Get No) Satisfaction (Mono)	Rock	1965
The Rolling Stones	Hot Rocks 1964-1971	Paint It Black	Rock	1966
Nina Simone	Wild Is The Wind	Black Is The Color Of My True Love's Hair (Live)	Джаз	1966
Bob Dylan	Bringing It All Back Home	Mr. Tambourine Man	Авторы-исполнители	1965
Bob Dylan	Bringing It All Back Home	It's All Over Now, Baby Blue	Авторы-исполнители	1965
Bob Dylan	Blonde On Blonde	Just Like a Woman	Рок	1966
Nina Simone	Wild Is The Wind	Four Women	Джаз	1966
Bob Dylan	Another Side of Bob Dylan	My Back Pages	Рок	1964
The Yardbirds	Five Live Yardbirds (Bonus Track Version)	I Ain't Got You	Рок	1964
Nina Simone	I Put a Spell On You	Feeling Good	Джаз	1965
Bob Dylan	Blonde On Blonde	Most Likely You Go Your Way (And I'll Go Mine)	Рок	1966
Bob Dylan	The Times They Are A-Changin'	The Lonesome Death of Hattie Carroll	Рок	1964
Bob Dylan	The Times They Are A-Changin'	The Times They Are A-Changin'	Рок	1964
Bob Dylan	Highway 61 Revisited	Like a Rolling Stone	Рок	1965
Nina Simone	Broadway-Blues-Ballads	See-Line Woman	Джаз	1964
John Mayall & The Bluesbreakers	Blues Breakers with Eric Clapton	Double Crossing Time (Stereo)	Блюз	1966
Nina Simone	Broadway-Blues-Ballads	Don't Let Me Be Misunderstood	Джаз	1964
Bob Dylan	Another Side of Bob Dylan	It Ain't Me Babe	Рок	1964
Bob Dylan	Bob Dylan's Greatest Hits	Positively 4th Street	Рок	1965
Bob Dylan	Blonde On Blonde	Rainy Day Women #12 & 35	Рок	1966
Bob Dylan	Highway 61 Revisited	Ballad of a Thin Man	Рок	1965
Bob Dylan	Highway 61 Revisited	Desolation Row	Рок	1965
Bob Dylan	Bringing It All Back Home	She Belongs to Me	Авторы-исполнители	1965
Bob Dylan	Highway 61 Revisited	Highway 61 Revisited	Рок	1965
Bob Dylan	Bringing It All Back Home	Subterranean Homesick Blues	Авторы-исполнители	1965
Bob Dylan	Bringing It All Back Home	Maggie's Farm	Авторы-исполнители	1965
Bob Dylan	Blonde On Blonde	Stuck Inside of Mobile with the Memphis Blues Again	Рок	1966
Bob Dylan	Blonde On Blonde	Visions of Johanna	Рок	1966
The Animals	Animalisms	Inside Looking Out (Bonus Track)	Рок	1966
The Animals	Animalisms	Don't Bring Me Down (Bonus Track)	Рок	1966
The Animals	Most of the Animals	We've Gotta Get Out of This Place	Поп	1965
The Animals	Most of the Animals	Bring It On Home to Me	Поп	1966
Nina Simone	Four Women: The Nina Simone Philips Recordings	Mississippi Goddam (Live In New York/1964)	Джаз	1964
Nina Simone	Four Women: The Nina Simone Philips Recordings	Pirate Jenny (Live At Carnegie Hall, New York/1964)	Джаз	1964
Nina Simone	Pastel Blues	Strange Fruit	Jazz	1965
Nina Simone	Pastel Blues	Be My Husband	Jazz	1965
Nina Simone	Let It All Out	The Other Woman (Live In New York/1964)	Jazz	1966
The Animals	The House of the Rising Sun: Best of the Animals	Gonna Send You Back to Walker	Pop	1964
The Animals	Most of the Animals	I'm Crying	Pop	1965
The Animals	Most of the Animals	Baby Let Me Take You Home	Pop	1964
The Animals	Animalisms	Boom Boom (Bonus Track)	Rock	1965
The Animals	Animalisms	Help Me Girl (Stereo)	Rock	1966
Nina Simone	Gospel Sunday	Sinnerman	Jazz	1965

6. GENERATION - Програма пропонує користувачу згенерувати дані для таблиці бази даних використовуючи таблицю з пропованих. Всі обмеження бази даних не порушуються протягом генерування даних.

```
Enter the action Command: 6
Enter table name from available ['artist' 'album' 'track' 'genre']: genre
Enter rows number to generate: 10
INSERT 0 10
Execution time: 0.012482166290283203 seconds
[INFO] Do you want to continue? [Y/n]
```

Результат генерування

```
| 57 | Country |
| 58 | Jazz |
| 59 | Classical |
| 60 | Rap |
| 61 | Disco |
| 62 | Easy Listening |
| 63 | Adult Alternative |
| 64 | Funk |
| 65 | TUIVFWJ |
| 66 | DVEELJT |
| 67 | WQIMEQE |
| 68 | TDUA0EA |
| 69 | STXJSGA |
| 70 | QGGVTSH |
| 71 | IYBVCDU |
| 72 | ICEJ0XJ |
| 73 | KIPQQSX |
| 74 | OGTQAES |
+-----+-----+
Execution time: 0.0001766681671142578 seconds
[INFO] Do you want to continue? [Y/n]
```

Завдання 1

Запит на видалення

Працюємо з таблицею жанр та трек. Кожен трек має свій жанр та посилається на таблицю жанр.

id	title	len	year	number_within_album	album_id	genre_id
1	Fortunate Son	00:00:12	1969	9	1	1
2	Lodi	00:00:31	1969	8	1	2
3	Run Through the Jungle	00:03:06	1970	11	1	1
4	I Put a Spell On You	00:18:56	1968	12	1	2
5	It Came Out of the Sky	00:05:59	1969	15	1	2
6	Susie Q	00:18:49	1968	13	1	2
7	Born On the Bayou	00:08:52	1969	2	1	2
8	Sweet Hitch-Hiker	00:17:42	1971	14	1	2
9	Who'll Stop the Rain	00:01:52	1970	16	1	2
10	Bad Moon Rising	00:06:41	1969	1	1	2
11	I Heard It Through the Grapevine (Edit)	00:18:38	1975	17	1	2
12	Molina	00:10:45	1970	21	1	2
13	Cotton Fields	00:16:20	1969	19	1	2
14	Hey Tonight	00:02:47	1970	18	1	2
15	Hello Mary Lou	00:03:04	1972	22	1	2
16	The Midnight Special	00:11:42	1969	23	1	2
17	Proud Mary	00:01:12	1968	3	1	2
18	Up Around the Bend	00:06:04	1970	24	1	2
19	Long As I Can See the Light	00:07:07	1970	20	1	2
20	Travelin' Band	00:12:19	1970	4	1	2
21	Lookin' Out My Back Door	00:14:12	1970	5	2	2
22	Have You Ever Seen the Rain	00:17:35	1968	5	1	2
23	Green River	00:02:49	1969	6	1	2
24	Down On the Corner	00:08:22	1968	7	1	2
25	Сансара (feat. Диана Арбенина, Александр Ф. Скляр, Сергей Бобунец, Sunsay, Скриптонит & Ант)	00:12:44	2017	1	3	3
26	Don't Cry (Original)	00:04:54	1991	4	4	4
27	November Rain	00:07:30	1991	10	4	4
28	Sympathy for the Devil	00:13:54	1994	15	5	4
29	Since I Don't Have You	00:11:10	1993	1	6	4
30	You Could Be Mine	00:02:53	1991	12	7	4
31	Civil War	00:13:28	1991	1	7	4
32	Paradise City	00:15:14	1987	6	8	4
33	Patience	00:10:19	1988	7	8	4
34	Ain't It Fun	00:13:24	1993	6	6	4
35	Knockin' On Heaven's Door	00:05:50	1991	4	7	4
36	Welcome to the Jungle	00:01:18	1987	1	8	4
37	Live and Let Die	00:14:42	1991	3	4	4

Ось перші декілька записів з таблиці трек. Деякі з них мають жанр під номером 4 , що має назву Hard Rock. Видалимо його

```
+-----+-----+
| Action      | Command |
+-----+-----+
| INSERT      | 1       |
| SELECT      | 2       |
| UPDATE      | 3       |
| DELETE      | 4       |
| COMPREHENSIVE | 5       |
| GENERATION  | 6       |
+-----+-----+

Enter the action Command: 4
Enter table name from available ['artist' 'album' 'track' 'genre']: genre
Enter attribute name from available ('id', 'name'),
if you want to delete row[s] by value in column.
If not -> press 'Enter' otherwise: id
Enter value to delete by: 4
DELETE 1
Execution time: 0.001171112060546875 seconds
[INFO] Do you want to continue? [Y/n]
```

Тепер подивимося таблицю трек.

id	title	len	year	number_within_album	album_id	genre_id
1	Fortunate Son	00:00:12	1969	9	1	1
2	Lodi	00:00:31	1969	8	1	2
3	Run Through the Jungle	00:03:06	1970	11	1	1
4	I Put a Spell On You	00:18:56	1968	12	1	2
5	It Came Out of the Sky	00:05:59	1969	15	1	2
6	Susie Q	00:18:49	1968	13	1	2
7	Born On the Bayou	00:08:52	1969	2	1	2
8	Sweet Hitch-Hiker	00:17:42	1971	14	1	2
9	Who'll Stop the Rain	00:01:52	1970	16	1	2
10	Bad Moon Rising	00:06:41	1969	1	1	2
11	I Heard It Through the Grapevine (Edit)	00:18:38	1975	17	1	2
12	Molina	00:10:45	1970	21	1	2
13	Cotton Fields	00:16:20	1969	19	1	2
14	Hey Tonight	00:02:47	1970	18	1	2
15	Hello Mary Lou	00:03:04	1972	22	1	2
16	The Midnight Special	00:11:42	1969	23	1	2
17	Proud Mary	00:01:12	1968	3	1	2
18	Up Around the Bend	00:06:04	1970	24	1	2
19	Long As I Can See the Light	00:07:07	1970	20	1	2
20	Travelin' Band	00:12:19	1970	4	1	2
21	Lookin' Out My Back Door	00:14:12	1970	5	2	2
22	Have You Ever Seen the Rain	00:17:35	1968	5	1	2
23	Green River	00:02:49	1969	6	1	2
24	Down On the Corner	00:08:22	1968	7	1	2
25	Сансара (feat. Диана Арбенина, Александр Ф. Скляр, Сергей Бобунец, Sunsay, Скриптонит & Ант)	00:12:44	2017	1	3	3
40	Неважно (Acoustic)	00:19:16	2013	16	9	5
41	Неважно	00:08:01	2013	10	9	5
42	Тралики	00:08:59	2013	6	9	5
43	Эндорфин	00:19:56	2013	11	9	5
44	Мотылёк	00:08:35	2013	3	9	5
45	Таёт дым (DJ Nejtrino & DJ Baur Remix)	00:10:48	2013	17	9	5
46	Рок	00:05:55	2003	8	8	5

Як ми бачимо, деякі записи зникли, це видно, зі значення айді рядків.

Цей функціонал як і у випадку з командою UPDATE, реалізовано на рівні бази даних завдяки обмеженням типу CASCADE.

Запит на вставку поля

Команда:

```
Enter the action Command: 1
Enter table name from available ['artist' 'album' 'track' 'genre']: track
Enter attribute[s] name from available ('id', 'title', 'len', 'year', 'number_within_album', 'album_id', 'genre_id'),
to insert data within the entered column[s]. If you want to insert values by several attributes -> separate attributes by comma ',': ti
tle, len, year, number_within_album, album_id, genre_id
Enter attribute[s] name from available ('id', 'title', 'len', 'year', 'number_within_album', 'album_id', 'genre_id'),
to insert data within the entered column[s]. If you want to insert values by several attributes -> separate values by comma ',': insert
ed, 00:59:59, 3000, 2, 233, 80
```

Результат:

```
[ERROR] insert or update on table "track" violates foreign key constraint "fk_genre_id"  
DETAIL:  Key (genre_id)=(80) is not present in table "genre".
```

Action	Command
INSERT	1
SELECT	2
UPDATE	3
DELETE	4
COMPREHENSIVE	5
GENERATION	6

Enter the action Command:

Тому що, жанру з айді 80 неіснує в таблиці жанрів.

Завдання 2

Вставка 5 псевдорандомізованих записів у кожен з таблиць.

1. artist

```
+-----+-----+
| Action      | Command |
+-----+-----+
| INSERT      | 1       |
| SELECT      | 2       |
| UPDATE      | 3       |
| DELETE      | 4       |
| COMPREHENSIVE | 5       |
| GENERATION  | 6       |
+-----+-----+
Enter the action Command: 2
Enter table name from available ['artist' 'album' 'track' 'genre']: artist
Enter attribute[s] name from available ('id', 'name'),
if you want to select particular column[s]. If not -> press 'Enter' otherwise:
+-----+-----+
| id | name |
+-----+-----+
| 1 | Creedence Clearwater Revival |
| 2 | Баста |
| 3 | Guns N' Roses |
| 4 | Макс Корж |
| 5 | Христина Соловій |
| 6 | MORGENSHTERN & Витя АК |
| 7 | Modern Talking |
| 8 | Комбинация |
| 9 | Quest Pistols |
| 10 | Usher |
| 11 | A2M |
| 12 | YOYU |
| 13 | Miyagi |
| 14 | Lynyrd Skynyrd |
| 15 | Ricchi e Poveri |
| 16 | Иван Дорн |
| 17 | Geri Halliwell |
| 18 | Король и Шут |
| 19 | КОРРУПЦИЯ |
| 20 | Любэ |
+-----+-----+
Execution time: 0.00014591217041015625 seconds
[INFO] Do you want to continue? [Y/n]
```

Змінена таблиця

```
Enter rows number to generate: 5
INSERT 0 5
Execution time: 0.006140232086181641 seconds
[INFO] Do you want to continue? [Y/n] Y
+-----+-----+
| Action      | Command |
+-----+-----+
| INSERT      | 1       |
| SELECT      | 2       |
| UPDATE      | 3       |
| DELETE      | 4       |
| COMPREHENSIVE | 5       |
| GENERATION  | 6       |
+-----+-----+
Enter the action Command: 2
Enter table name from available ['artist' 'album' 'track' 'genre']: artist
Enter attribute[s] name from available ('id', 'name'),
if you want to select particular column[s]. If not -> press 'Enter' otherwise:
+-----+-----+
| id | name |
+-----+-----+
| 1 | Creedence Clearwater Revival |
| 2 | Баста |
| 3 | Guns N' Roses |
| 4 | Макс Корж |
| 5 | Христина Соловій |
| 6 | MORGENSHTERN & Витя АК |
| 7 | Modern Talking |
| 8 | Комбинация |
| 9 | Quest Pistols |
| 10 | Usher |
| 11 | A2M |
| 12 | YOFU |
| 13 | Miyagi |
| 14 | Lynyrd Skynyrd |
| 15 | Ricchi e Poveri |
| 16 | Иван Дорн |
| 17 | Geri Halliwell |
| 18 | Король и Шут |
| 19 | КОРРУПЦИЯ |
| 20 | Любэ |
| 902 | OUYAPW |
| 903 | YYTSKDH |
| 904 | NWYCHRM |
| 905 | FFKQMXU |
| 906 | YGYNVMF |
```


******При генеруванні даних, sequence таблиці не оновлюється, тому не маємо послідовних айді у генерованих даних, відповідно існуючим.

2. Genre

```
Enter the action Command: 2
Enter table name from available ['artist' 'album' 'track' 'genre']: genre
Enter attribute[s] name from available ('id', 'name'),
if you want to select particular column[s]. If not -> press 'Enter' otherwise:
+-----+-----+
|  id | name      |
|-----+-----|
|   1 | Rock      |
|   2 | Рок       |
|   3 | Hip-Hop/Rap |
|   5 | Поп       |
+-----+-----+
Execution time: 0.0001399517059326172 seconds
[INFO] Do you want to continue? [Y/n] Y
```

Змінена таблиця

```
+-----+-----+
| Action      | Command |
+-----+-----+
| INSERT      | 1 |
| SELECT      | 2 |
| UPDATE      | 3 |
| DELETE      | 4 |
| COMPREHENSIVE | 5 |
| GENERATION   | 6 |
+-----+-----+
Enter the action Command: 6
Enter table name from available ['artist' 'album' 'track' 'genre']: genre
Enter rows number to generate: 5
INSERT 0 5
Execution time: 0.009427309036254883 seconds
[INFO] Do you want to continue? [Y/n] Y
+-----+-----+
| Action      | Command |
+-----+-----+
| INSERT      | 1 |
| SELECT      | 2 |
| UPDATE      | 3 |
| DELETE      | 4 |
| COMPREHENSIVE | 5 |
| GENERATION   | 6 |
+-----+-----+
Enter the action Command: 2
Enter table name from available ['artist' 'album' 'track' 'genre']: genre
Enter attribute[s] name from available ('id', 'name'),
if you want to select particular column[s]. If not -> press 'Enter' otherwise:
+-----+-----+
| id | name      |
+-----+-----+
| 1 | Rock      |
| 2 | Рок      |
| 3 | Hip-Hop/Rap |
| 5 | Поп      |
| 70 | UDORRBQ   |
| 71 | XMCOCIG   |
| 72 | BORCONV   |
| 73 | JQJCFWS   |
| 74 | QJEBTFX   |
+-----+-----+
Execution time: 0.00016069412231445312 seconds
[INFO] Do you want to continue? [Y/n]
```

3. Album

```
+-----+-----+
| Action      | Command |
+-----+-----+
| INSERT      | 1 |
| SELECT      | 2 |
| UPDATE      | 3 |
| DELETE      | 4 |
| COMPREHENSIVE | 5 |
| GENERATION  | 6 |
+-----+-----+

Enter the action Command: 2
Enter table name from available ['artist' 'album' 'track' 'genre']: album
Enter attribute[s] name from available ('id', 'title', 'tracks_number', 'artist_id'),
if you want to select particular column[s]. If not -> press 'Enter' otherwise:

+-----+-----+-----+-----+
| id | title | tracks_number | artist_id |
+-----+-----+-----+-----+
| 1 | Best of Creedence Clearwater Revival | 24 | 1 |
| 2 | Cosmo's Factory | 11 | 1 |
| 3 | Сансара (feat. Диана Арбенина, Александр Ф. Скляр, Сергей Бобунец, Sunsay, Скриптонит & Ант) - Single | 1 | 2 |
| 4 | Use Your Illusion I | 16 | 3 |
| 5 | Greatest Hits (Bonus Track Version) | 15 | 3 |
+-----+-----+-----+-----+

Execution time: 0.000156402587890625 seconds
[INFO] Do you want to continue? [Y/n]
```

Змінена таблиця

```
+-----+
| Action      | Command |
+-----+
| INSERT      | 1 |
| SELECT      | 2 |
| UPDATE      | 3 |
| DELETE      | 4 |
| COMPREHENSIVE | 5 |
| GENERATION  | 6 |
+-----+

Enter the action Command: 6
Enter table name from available ['artist' 'album' 'track' 'genre']: album
Enter rows number to generate: 5
INSERT 0 5
Execution time: 0.004976749420166016 seconds
[INFO] Do you want to continue? [Y/n] Y

+-----+
| Action      | Command |
+-----+
| INSERT      | 1 |
| SELECT      | 2 |
| UPDATE      | 3 |
| DELETE      | 4 |
| COMPREHENSIVE | 5 |
| GENERATION  | 6 |
+-----+

Enter the action Command: 2
Enter table name from available ['artist' 'album' 'track' 'genre']: album
Enter attribute[s] name from available ('id', 'title', 'tracks_number', 'artist_id'),
if you want to select particular column[s]. If not -> press 'Enter' otherwise:

+-----+-----+-----+
| id | title | tracks_number | artist_id |
+-----+-----+-----+
| 1 | Best of Creedence Clearwater Revival | 24 | 1 |
| 2 | Cosmo's Factory | 11 | 1 |
| 3 | Сансара (feat. Диана Арбенина, Александр Ф. Скляр, Сергей Бобунец, Sunsay, Скриптонит & Ант) - Single | 1 | 2 |
| 4 | Use Your Illusion I | 16 | 3 |
| 5 | Greatest Hits (Bonus Track Version) | 15 | 3 |
| 1759 | KFTUDHY | 8 | 14 |
| 1760 | SHLEADD | 22 | 905 |
| 1761 | ALSETNI | 12 | 13 |
| 1762 | IGPLIWY | 16 | 3 |
| 1763 | VMNBIHV | 49 | 20 |
+-----+-----+-----+
```

**** Обмеження з зовнішніх ключів дотримуються. Обмеження на унікальність значення також для генерації усіх таблиць – враховується. Подивимось таблицю артист.**

Action	Command
INSERT	1
SELECT	2
UPDATE	3
DELETE	4
COMPREHENSIVE	5
GENERATION	6

Enter the action Command: 2

Enter table name from available ['artist' 'album' 'track' 'genre']: artist

Enter attribute[s] name from available ('id', 'name'),

if you want to select particular column[s]. If not -> press 'Enter' otherwise:

id	name
1	Creedence Clearwater Revival
2	Баста
3	Guns N' Roses
4	Макс Корж
5	Христина Соловій
6	MORGENSHTERN & Витя АК
7	Modern Talking
8	Комбинация
9	Quest Pistols
10	Usher
11	A2M
12	YOFU
13	Miyagi
14	Lynyrd Skynyrd
15	Ricchi e Poveri
16	Иван Дорн
17	Geri Halliwell
18	Король и Шут
19	КОРРУПЦИЯ
20	Любэ
902	OYWYAPW
903	YYTSKDH
904	NWYCHRM
905	FFKQMXU
906	YGYNVMF

Execution time: 0.0001537799835205078 seconds

[INFO] Do you want to continue? [Y/n]

4. Track

```
+-----+
| Action      | Command |
+-----+
| INSERT      | 1 |
| SELECT      | 2 |
| UPDATE      | 3 |
| DELETE      | 4 |
| COMPREHENSIVE | 5 |
| GENERATION  | 6 |
+-----+

Enter the action Command: 2
Enter table name from available ['artist' 'album' 'track' 'genre']: track
Enter attribute[s] name from available ('id', 'title', 'len', 'year', 'number_within_album', 'album_id', 'genre_id'),
if you want to select particular column[s]. If not -> press 'Enter' otherwise:

+-----+-----+-----+-----+-----+-----+
| id | title | len | year | number_within_album | album_id | genre_id |
+-----+-----+-----+-----+-----+-----+
| 1 | Fortunate Son | 00:00:12 | 1969 | 9 | 1 | 1 |
| 2 | Lodi | 00:00:31 | 1969 | 8 | 1 | 2 |
| 3 | Run Through the Jungle | 00:03:06 | 1970 | 11 | 1 | 1 |
| 4 | I Put a Spell On You | 00:18:56 | 1968 | 12 | 1 | 2 |
| 5 | It Came Out of the Sky | 00:05:59 | 1969 | 15 | 1 | 2 |
| 6 | Susie Q | 00:18:49 | 1968 | 13 | 1 | 2 |
| 7 | Born On the Bayou | 00:08:52 | 1969 | 2 | 1 | 2 |
| 8 | Sweet Hitch-Hiker | 00:17:42 | 1971 | 14 | 1 | 2 |
| 9 | Who'll Stop the Rain | 00:01:52 | 1970 | 16 | 1 | 2 |
| 10 | Bad Moon Rising | 00:06:41 | 1969 | 1 | 1 | 2 |
| 11 | I Heard It Through the Grapevine (Edit) | 00:18:38 | 1975 | 17 | 1 | 2 |
| 12 | Molina | 00:10:45 | 1970 | 21 | 1 | 2 |
| 13 | Cotton Fields | 00:16:20 | 1969 | 19 | 1 | 2 |
| 14 | Hey Tonight | 00:02:47 | 1970 | 18 | 1 | 2 |
| 15 | Hello Mary Lou | 00:03:04 | 1972 | 22 | 1 | 2 |
| 16 | The Midnight Special | 00:11:42 | 1969 | 23 | 1 | 2 |
| 17 | Proud Mary | 00:01:12 | 1968 | 3 | 1 | 2 |
| 18 | Up Around the Bend | 00:06:04 | 1970 | 24 | 1 | 2 |
| 19 | Long As I Can See the Light | 00:07:07 | 1970 | 20 | 1 | 2 |
| 20 | Travelin' Band | 00:12:19 | 1970 | 4 | 1 | 2 |
| 21 | Lookin' Out My Back Door | 00:14:12 | 1970 | 5 | 2 | 2 |
| 22 | Have You Ever Seen the Rain | 00:17:35 | 1968 | 5 | 1 | 2 |
| 23 | Green River | 00:02:49 | 1969 | 6 | 1 | 2 |
| 24 | Down On the Corner | 00:08:22 | 1968 | 7 | 1 | 2 |
| 25 | Сансара (feat. Диана Арбенина, Александр Ф. Скляр, Сергей Бобунец, Sunsay, Скриптонит & Ант) | 00:12:44 | 2017 | 1 | 3 | 3 |
+-----+-----+-----+-----+-----+-----+

Execution time: 0.0001723766326904297 seconds
[INFO] Do you want to continue? [Y/n]
```

Змінена таблиця

```
Execution time: 0.0001723766326904297 seconds
[INFO] Do you want to continue? [Y/n] Y

+-----+-----+
| Action      | Command |
+-----+-----+
| INSERT      | 1 |
| SELECT      | 2 |
| UPDATE      | 3 |
| DELETE      | 4 |
| COMPREHENSIVE | 5 |
| GENERATION  | 6 |
+-----+-----+

Enter the action Command: 6
Enter table name from available ['artist' 'album' 'track' 'genre']: track
Enter rows number to generate: 5
INSERT 0 5

Execution time: 0.0042836666107177734 seconds
[INFO] Do you want to continue? [Y/n] Y
```

Enter the action Command: 2
Enter table name from available ['artist' 'album' 'track' 'genre']: track
Enter attribute[s] name from available ('id', 'title', 'len', 'year', 'number_within_album', 'album_id', 'genre_id'),
if you want to select particular column[s]. If not -> press 'Enter' otherwise:

id	title	len	year	number_within_album	album_id	genre_id
1	Fortunate Son	00:00:12	1969	9	1	1
2	Lodi	00:00:31	1969	8	1	2
3	Run Through the Jungle	00:03:06	1970	11	1	1
4	I Put a Spell On You	00:18:56	1968	12	1	2
5	It Came Out of the Sky	00:05:59	1969	15	1	2
6	Susie Q	00:18:49	1968	13	1	2
7	Born On the Bayou	00:08:52	1969	2	1	2
8	Sweet Hitch-Hiker	00:17:42	1971	14	1	2
9	Who'll Stop the Rain	00:01:52	1970	16	1	2
10	Bad Moon Rising	00:06:41	1969	1	1	2
11	I Heard It Through the Grapevine (Edit)	00:18:38	1975	17	1	2
12	Molina	00:10:45	1970	21	1	2
13	Cotton Fields	00:16:20	1969	19	1	2
14	Hey Tonight	00:02:47	1970	18	1	2
15	Hello Mary Lou	00:03:04	1972	22	1	2
16	The Midnight Special	00:11:42	1969	23	1	2
17	Proud Mary	00:01:12	1968	3	1	2
18	Up Around the Bend	00:06:04	1970	24	1	2
19	Long As I Can See the Light	00:07:07	1970	20	1	2
20	Travelin' Band	00:12:19	1970	4	1	2
21	Lookin' Out My Back Door	00:14:12	1970	5	2	2
22	Have You Ever Seen the Rain	00:17:35	1968	5	1	2
23	Green River	00:02:49	1969	6	1	2
24	Down On the Corner	00:08:22	1968	7	1	2
25	Сансара (feat. Диана Арбенина, Александр Ф. Скляр, Сергей Бобунец, Sunsay, Скриптонит & Ант)	00:12:44	2017	1	3	3
3453	BIWQWQO	10:30:00	1944	15	1762	5
3454	OGFAXEW	22:32:56	2020	9	2	5
3455	WJVCYQC	20:53:06	1950	2	1761	71
3456	YQSPUSW	00:21:44	1955	6	2	73
3457	JKHEXKS	20:11:29	1956	11	1761	1

Execution time: 0.00018262863159179688 seconds
[INFO] Do you want to continue? [Y/n]

Порушення цілостності зовнішніх ключів немає.

Реалізація

Генерація виконується не завжди одним sql запитом, тому що в такому випадку, була б дуже велика кількість однокового коду. Таким чином було прийнято рішення оптимізувати процес за допомогою пайтон. *Усі дані, що, в результаті вставляються в таблицю згенеровані тільки sql запитом.

```
self.text_request = f"SELECT chr(trunc(65 + random()*25)::int) || " \
                    f"chr(trunc(65 + random()*25)::int) || " \
                    f"chr(trunc(65 + random()*25)::int) || " \
                    f"chr(trunc(65 + random()*25)::int) || " \
                    f"from generate_series(1, 1)"
```

Оголошено змінну об'єкту класу, задля не дублювання потім одного й того ж самого sql запиту.

Ці дві функції генерують значення для таблиць artist, genre. Так як, вони

```
def get_if_unique(self, table: str, attr: str) -> bool:
    with self.model.conn.cursor() as cursor:
        sql_get_all = f"SELECT {attr} FROM {table};"
        cursor.execute(sql_get_all)
        elements = tuple(element[0] for element in cursor.fetchall())
        cursor.execute(self.text_request)
        value = cursor.fetchall()[0][0]
        if value in elements:
            return False
        return value

def generate_one_unique_attr_table(self, table: str, rows_to_generate: int):
    try:
        n_gen_rows = 0
        with self.model.conn.cursor() as cursor:
            while n_gen_rows != rows_to_generate:
                value = self.get_if_unique(table, "name")
                if value:
                    sql_insert = f"INSERT INTO {table} (name) VALUES (\'{value}\')"
                    cursor.execute(sql_insert)
                    n_gen_rows += 1
            self.model.conn.commit()
```

складаються тільки з одного неключового атрибуту – їх генерація может будети автоматизована мовою програмування.

Таким чином виглядає уся структура роботи функції генерування. Тут також надано sql запити для генерування таблиць альбом та трек.

```
n_gen_rows = 0
start = time()
if "genre" == table:
    self.generate_one_unique_attr_table("genre", rows_to_generate)
elif "artist" == table:
    self.generate_one_unique_attr_table("artist", rows_to_generate)
elif "album" == table:
    while n_gen_rows != rows_to_generate:

        value = self.get_if_unique(table, "title")
        if not value:
            continue
        sql_insert = f"INSERT INTO album (title, tracks_number, artist_id) VALUES (" \
            f"({self.text_request}), " \
            f"(SELECT trunc(random()*(50-1)+1)::int FROM generate_series(1, 1)), " \
            f"(SELECT id FROM artist ORDER BY RANDOM() LIMIT 1))"
        cursor.execute(sql_insert)
        n_gen_rows += 1
elif "track" == table:
    while n_gen_rows != rows_to_generate:
        sql_album_id_tracks_num = "SELECT id, tracks_number FROM album ORDER BY RANDOM() LIMIT 1"
        cursor.execute(sql_album_id_tracks_num)
        album_id, album_tracks = cursor.fetchall()[0]
        sql_insert = f"INSERT INTO track (title, len, year, number_within_album, album_id, genre_id) " \
            f"VALUES (" \
            f"({self.text_request}), " \
            f"(SELECT chr(trunc(48+random()*3)::int) || chr(trunc(48+random()*4)::int) || ':' " \
            f"|| chr(trunc(48+random()*6)::int) || chr(trunc(48+random()*10)::int) || ':' " \
            f"|| chr(trunc(48+random()*6)::int) || chr(trunc(48+random()*10)::int) " \
            f"FROM generate_series(1, 1)), " \
            f"(SELECT trunc(random()*(2021-1940+1)+1940)::int FROM generate_series(1, 1)), " \
            f"(SELECT trunc(floor(random()*({album_tracks}-1+1)+1)::int " \
            f"FROM generate_series(1, 1)), " \
            f"{album_id}, " \
            f"(SELECT id FROM genre ORDER BY random() LIMIT 1));"
        cursor.execute(sql_insert)
        n_gen_rows += 1

result = time() - start
self.model.conn.commit()
return f"INSERT 0 {rows_to_generate}", result
except Exception as _ex:
    raise ex_hadler.Error(_ex)
```

Завдання 3. Складні sql запити

1. пошук в рамках діапазону років.

```
Enter the action Command: 5
Choose from available modes ('SELECT',): SELECT
Choose from available options ('year range', 'length range', 'album size'): year range
Enter minimal and maximum publishing years, separated by space

1980 1990

+-----+-----+-----+-----+-----+
| Group Name | Album Title | Song Title | Genre | Year |
+-----+-----+-----+-----+-----+
| Guns N' Roses | Appetite for Destruction (Super Deluxe) | Paradise City | Hard Rock | 1987 |
| Guns N' Roses | Appetite for Destruction (Super Deluxe) | Patience | Hard Rock | 1988 |
| Guns N' Roses | Appetite for Destruction (Super Deluxe) | Welcome to the Jungle | Hard Rock | 1987 |
| Guns N' Roses | Appetite for Destruction (Super Deluxe) | Sweet Child O' Mine | Hard Rock | 1987 |
| Modern Talking | 25 Years of Disco-Pop | Cheri Cheri Lady | Pop | 1985 |
| Modern Talking | 25 Years of Disco-Pop | Brother Louie | Pop | 1986 |
| Guns N' Roses | Appetite for Destruction (Super Deluxe) | You're Crazy | Хард-рок | 1988 |
| Guns N' Roses | Appetite for Destruction (Super Deluxe) | Used to Love Her | Хард-рок | 1988 |
| Guns N' Roses | Appetite for Destruction | Think About You | Hard Rock | 1987 |
| Guns N' Roses | Appetite for Destruction | My Michelle | Hard Rock | 1987 |
| Guns N' Roses | Appetite for Destruction | Mr. Brownstone | Hard Rock | 1987 |
| Guns N' Roses | Appetite for Destruction | Out ta Get Me | Hard Rock | 1987 |
| Guns N' Roses | Appetite for Destruction | Nightrain | Hard Rock | 1987 |
| Guns N' Roses | Appetite for Destruction | It's So Easy | Hard Rock | 1987 |
| Guns N' Roses | Appetite for Destruction | Rocket Queen | Hard Rock | 1987 |
+-----+-----+-----+-----+-----+
Execution time: 0.0008039474487304688 seconds
[INFO] Do you want to continue? [Y/n]
```

```
sql = f"""select artist.name as "Group Name",
           album.title as "Album Title",
           track.title as "Song Title",
           genre.name as "Genre",
           track.year as "Year"
from artist inner join album
on album.artist_id = artist.id inner join track
on track.album_id = album.id inner join genre
on track.genre_id = genre.id
where track.year >= {min_year}
and track.year <= {max_year};

"""
```

2. Пошук в рамках діапазону тривалості пісні.

```
Enter the action Command: 5
Choose from available modes ('SELECT',): SELECT
Choose from available options ('year range', 'length range', 'album size'): length range
Enter minimal and maximum song duration , in format 00:00:00, separated by space

00:03:00 00:05:00

+-----+-----+-----+-----+-----+
| Album Title | Song Title | Song Duration | Number in Album | Alsum Size |
+-----+-----+-----+-----+-----+
| Best of Creedence Clearwater Revival | Run Through the Jungle | 00:03:32 | 11 | 24 |
| Best of Creedence Clearwater Revival | Down On the Corner | 00:04:01 | 7 | 24 |
| Appetite for Destruction (Super Deluxe) | Welcome to the Jungle | 00:03:33 | 1 | 12 |
| Жить в кайф | Таёт дым (DJ Nejtrino & DJ Baur Remix) | 00:03:48 | 17 | 17 |
| Жить в кайф | Нет никаких правил | 00:04:35 | 14 | 17 |
| Малый повзрослел, Ч. 2 | Молодость всё прощает | 00:04:32 | 2 | 7 |
| Животный мир | Армия | 00:03:21 | 9 | 18 |
| Животный мир | Небо поможет нам (feat. Sasha Plus) [Remix] | 00:03:20 | 18 | 18 |
| All the Best | Come vorrei | 00:04:44 | 7 | 12 |
| Use Your Illusion I | Coma | 00:03:42 | 16 | 16 |
| Chinese Democracy | Chinese Democracy | 00:04:58 | 1 | 14 |
| Театр дэмона | Послание | 00:03:27 | 1 | 12 |
| Герои и злодеи | Кузнец | 00:04:25 | 7 | 11 |
| Герои и злодеи | Охотник | 00:03:38 | 1 | 11 |
| Герои и злодеи | Медведь | 00:04:24 | 2 | 11 |
| Продавец кошмаров | Пляски на могиле | 00:03:32 | 13 | 17 |
| Продавец кошмаров | Гробовщик | 00:04:01 | 4 | 17 |
| Продавец кошмаров | Продавец кошмаров | 00:03:51 | 15 | 17 |
| Жаль, нет ружья! | Пьянка | 00:04:24 | 17 | 17 |
| Жаль, нет ружья! | Утопленник | 00:03:00 | 15 | 17 |
| Как в старой сказке | Скотный двор | 00:03:01 | 6 | 17 |
| Акустический альбом | Карапуз | 00:04:35 | 10 | 21 |
| Акустический альбом | Прыгну со скалы (Bonus Version) | 00:03:44 | 21 | 21 |
| Акустический альбом | Молотов (Instrumental) | 00:04:29 | 19 | 21 |
| Акустический альбом | Ведьма и осёл | 00:04:00 | 12 | 21 |
| 40 | Поколение X (feat. ANIKV & Noize MC) | 00:03:12 | 11 | 23 |
| Мы с тобой (feat. Алсу) - Single | Мы с тобой (feat. Алсу) | 00:04:42 | 1 | 1 |
| Баста 3 | Ростов | 00:04:02 | 4 | 20 |
| Appetite for Destruction | Nightrain | 00:03:58 | 3 | 12 |
+-----+-----+-----+-----+-----+

Execution time: 0.0013184547424316406 seconds
★ [INFO] Do you want to continue? [Y/n]
```

```
sql = f"""select album.title as "Album Title",
            track.title as "Song Title",
            track.len as "Song Duration",
            track.number_within_album as "Number in Album",
            album.tracks_number as "Alsum Size"
from album inner join track
on album.id = track.album_id
where track.len <= \'{max_length}\''
and track.len >= \'{min_length}\'';
```

3. Пошук в діапазоні кількості пісень, які складають альбом

```
Terminal: Local x Local(2) x + v
Enter the action Command: 5
Choose from available modes ('SELECT',): SELECT
Choose from available options ('year range', 'length range', 'album size'): album size
Enter minimal and maximum song number of songs in album, separated by space

10 20
+-----+-----+-----+
| Group Name           | Album Title                               | Album Size |
+-----+-----+-----+
| Creedence Clearwater Revival | Cosmo's Factory                          | 11 |
| Guns N' Roses          | Use Your Illusion I                      | 16 |
| Guns N' Roses          | Greatest Hits (Bonus Track Version)     | 15 |
| Guns N' Roses          | The Spaghetti Incident?                 | 12 |
| Guns N' Roses          | Use Your Illusion II                    | 14 |
| Guns N' Roses          | Appetite for Destruction (Super Deluxe) | 12 |
| Макс Корж              | Жизнь в кайф                             | 17 |
| Макс Корж              | Малый повзрослел, Ч. 1                  | 10 |
| Макс Корж              | Животный мир                             | 18 |
| Христина Соловій       | Жива вода                               | 12 |
| MORGENSHTERN & Витя АК | ЛЕГЕНДАРНАЯ ПЫЛЬ                        | 10 |
| Modern Talking         | 25 Years of Disco-Pop                   | 17 |
| Комбинация             | Легендарные песни                       | 15 |
| Quest Pistols          | Волшебные краски                        | 16 |
| Miyagi                 | Buster Keaton                           | 12 |
| Ricchi e Poveri        | All the Best                            | 12 |
| Иван Дорн              | CO'N'DORN                               | 16 |
| Geri Halliwell         | Scream If You Wanna Go Faster           | 12 |
| Король и Шут           | Герои и злодеи                         | 11 |
| КОРРУПЦИЯ              | Журавли                                 | 10 |
| Король и Шут           | Король и Шут (Будь как дома, путник...) | 17 |
| Иван Дорн              | ОТД                                      | 12 |
| Guns N' Roses          | Chinese Democracy                       | 14 |
| Король и Шут           | Как в старой сказке                     | 17 |
| Король и Шут           | Театр дэмона                           | 12 |
| Король и Шут           | Бунт на корабле                         | 15 |
| Король и Шут           | Продавец кошмаров                       | 17 |
| Король и Шут           | Жаль, нет ружья!                       | 17 |
| Баста                  | К тебе                                  | 12 |
| Баста                  | Баста 5, Часть 2                        | 12 |
| Баста                  | Баста 2                                 | 17 |
| Баста                  | Баста+                                  | 17 |
| Баста                  | Баста 4                                 | 17 |
| Баста                  | Баста 3                                 | 20 |
| Баста                  | Баста 1                                 | 19 |
| Guns N' Roses          | Appetite for Destruction                 | 12 |
+-----+-----+-----+
Execution time: 0.0005414485931396484 seconds
[INFO] Do you want to continue? [Y/n]
```

```
sql = f"""select artist.name as "Group Name",  
           album.title as "Album Title",  
           album.tracks_number as "Album Size"  
from artist inner join album  
on album.artist_id = artist.id  
where album.tracks_number <= {max_size}  
and album.tracks_number >= {min_size};"""
```


Завдання 4

Код програмного модулю “model.py”

```
import postgresql_backend
from typing import *
from numpy import array

class ModelPostgreSQL:
    def __init__(self):
        self.__connection = postgresql_backend.connect_to_db()

    @property
    def conn(self):
        return self.__connection

    @property
    def tables(self):
        with self.conn:
            with self.conn.cursor() as cursor:
                cursor.execute("""SELECT table_name FROM information_schema.tables
                                WHERE table_schema = 'public'""")
                return array(array(cursor.fetchall())[:, 0])

    def create(self, table: str, attrs: str | Tuple[str, ...], values: str | Tuple[str, ...]) -> Tuple[str, float]:
        return postgresql_backend.insert(self.conn, table, attrs, values)

    def read(self, table: str, attrs: str = None) -> Tuple[List[Tuple[str, ...]], Tuple[str, ...], float]:
        return postgresql_backend.select(self.conn, table, attrs)

    def update(self, table: str, set_attrs: str | Tuple[str, ...], set_values: str | Tuple[str, ...],
               where_attrs: str, where_values: str) -> Tuple[str, float]:
        return postgresql_backend.update(self.conn, table, set_attrs, set_values, where_attrs, where_values)

    def delete(self, table: str, attr: str, value: str) -> Tuple[str, float]:
        return postgresql_backend.delete(self.conn, table, attr, value)

    def disconnect(self):
        postgresql_backend.disconnect_from_db(self.conn)

    @staticmethod
    def get_table_columns(conn: psql_conn, table: str) -> tuple[str, ...]:
        return postgresql_backend.get_table_columns(conn, table)
```

Код программного модуля “postgresql_backend.py”

```
import psycopg2
import ex_hadler

from psycopg2 import ProgrammingError
from typing import *
from numpy import array
from time import time

def connect_to_db() -> psycopg2.Connection | None:
    try:
        connection = psycopg2.connect(
            database="music_test",
            user="postgres",
            password="*****",
            host="127.0.0.1"
        )

        with connection:
            with connection.cursor() as cursor:
                cursor.execute("SELECT version();")

        return connection

    except Exception as _ex:
        raise ex_hadler.Error(_ex)

def disconnect_from_db(conn: psycopg2.Connection) -> None:
    if conn is not None:
        conn.close()

def connect(func: Callable) -> Callable:
    def inner_func(conn: psycopg2.Connection, *args, **kwargs) -> Callable:
        try:
            with conn:
                with conn.cursor() as cursor:
                    cursor.execute("SELECT version();")
        except (AttributeError, ProgrammingError) as _ex:
            conn = connect_to_db()
        return func(conn, *args, **kwargs)
    return inner_func

@connect
def insert(conn: psycopg2.Connection, table_name: str, attributes: str | Tuple[str, ...],
           values: str | Tuple[str, ...]) -> Tuple[str, float]:
    # single or multiple insert query handle
```

```

    sql = f"INSERT INTO {table_name} ({', '.join([attributes])}) VALUES
(\'{values}\') " \
    if type(values) is not tuple \
    else f"INSERT INTO {table_name} ({', '.join(attributes)}) VALUES ({',
'.join(['%s'] * len(values))}"
    try:
        with conn.cursor() as cursor:
            start = time()
            cursor.execute(sql, values)
            exec_time = time() - start
            status_msg = cursor.statusmessage
            conn.commit()
            return status_msg, exec_time
    except Exception as _ex:
        raise ex_handler.Error(_ex)

@connect
def select(conn: psql_conn, table_name: str, attributes: str | Tuple[str, ...] =
None) \
    -> Tuple[List[Tuple[str, ...]], Tuple[str, ...], float]:
    # selecting all table or specified rows
    base = f"SELECT * FROM {table_name}"
    if attributes is not None:
        # if attributed request -> check how many attributes are requested
        sql = f"SELECT {attributes} FROM {table_name}" \
            if type(attributes) is str \
            else f"SELECT ({', '.join(attributes)}) FROM {table_name}"
    else:
        sql = base

    try:
        with conn.cursor() as cursor:
            start = time()
            cursor.execute(sql)
            exec_time = time() - start
            conn.commit()
            return cursor.fetchall(), tuple(array(cursor.description)[: , 0]),
exec_time
    except Exception as _ex:
        raise ex_handler.Error(_ex)

# update on cascade handles parent-child relation
@connect
def update(conn: psql_conn, table_name: str,
           set_attrs: str | Tuple[str, ...],
           set_values: str | Tuple[str, ...],
           where_attr: str, where_value: str) -> tuple[str, float] | None:

    batch = True if type(set_attrs) is tuple else False # updating many
attributes or one

    sql = f"UPDATE {table_name} SET {' = %s, '.join(set_attrs) + ' = %s'} WHERE

```



```

{where_attr} = %s" \
    if batch \
        else f"UPDATE {table_name} SET {' = %s, '.join([set_attrs]) + ' = %s'}
WHERE {where_attr} = %s"

    try:
        with conn.cursor() as cursor:
            start = time()
            cursor.execute(sql, (set_values + (where_value,)) if batch \
                               else cursor.execute(sql, ((set_values,) + (where_value,))))
            exec_time = time() - start
            status_msg = cursor.statusmessage
            conn.commit()
            return status_msg, exec_time
    except Exception as _ex:
        raise ex_hadler.Error(_ex)

# delete on cascade handles parent-child relation
@connect
def delete(conn: psql_conn, table_name: str, attribute: str, value: str) ->
tuple[str, float] | None:
    sql = f"DELETE FROM {table_name} WHERE {attribute} = %s"

    try:
        with conn.cursor() as cursor:
            start = time()
            cursor.execute(sql, (value, ))
            exec_time = time() - start
            status_msg = cursor.statusmessage
            conn.commit()
            return status_msg, exec_time
    except Exception as _ex:
        raise ex_hadler.Error(_ex)

def get_table_columns(conn: psql_conn, table: str) -> Tuple[str]:
    sql = f"SELECT * FROM {table};"
    try:
        with conn.cursor() as cursor:
            cursor.execute(sql)
            conn.commit()
            return tuple(array(cursor.description)[: , 0])
    except Exception as _ex:
        raise ex_hadler.Error(_ex)

```

Завдання 2

```
class SQLGenerator(Analyzer):

    def __init__(self, model: ModelPostgreSQL, xml_source: str, is_empty: bool = False):
        super().__init__(model, is_empty)
        self.doc_path = xml_source
        self.text_request = f"SELECT chr(trunc(65 + random()*25)::int) || " \
                             f"chr(trunc(65 + random()*25)::int) || chr(trunc(65 + \
random()*25)::int) || " \
                             f"chr(trunc(65 + random()*25)::int) || chr(trunc(65 + \
random()*25)::int) || " \
                             f"chr(trunc(65 + random()*25)::int) || chr(trunc(65 + \
random()*25)::int) " \
                             f"from generate_series(1, 1)"

    def create_tables(self, conn: psql_conn): ...

    def get_if_unique(self, table: str, attr: str) -> bool:
        with self.model.conn.cursor() as cursor:
            sql_get_all = f"SELECT {attr} FROM {table};"
            cursor.execute(sql_get_all)
            elements = tuple(element[0] for element in cursor.fetchall())
            cursor.execute(self.text_request)
            value = cursor.fetchall()[0][0]
            if value in elements:
                return False
            return value

    def generate_one_unique_attr_table(self, table: str, rows_to_generate: int):
        try:
            n_gen_rows = 0
            with self.model.conn.cursor() as cursor:
                while n_gen_rows != rows_to_generate:
                    value = self.get_if_unique(table, "name")
                    if value:
                        sql_insert = f"INSERT INTO {table} (name) VALUES \
(\'{value}\')"
                        cursor.execute(sql_insert)
                        n_gen_rows += 1
                self.model.conn.commit()

        except Exception as _ex:
            raise ex_hadler.Error(_ex)

    def generate_table(self, table: str, rows_to_generate: int) -> Tuple[str, float]:
        table_analysis = self.type_analysis[table]
        if False in [is_valid for data_type, is_valid in table_analysis.values()]:
            raise TypeError("The required type is not supported for generation.")
        try:
```



```

        and track.year <= {max_year};

        """
    elif "length range" == option:
        while True:
            try:
                min_length, max_length = input(
                    f"Enter minimal and maximum song duration , in format
00:00:00, "
                    f"separated by space\n\n").split()
                if Validator.is_len_type(min_length) and
Validator.is_len_type(max_length):
                    break
                self.view.display_warning("Entered length is not in format
[0]00:00:00. Try again.")
            except Exception as _ex:
                raise ValueError("Is required 2 parameters. Try again.")

        sql = f"""select album.title as "Album Title",
                    track.title as "Song Title",
                    track.len as "Song Duration",
                    track.number_within_album as "Number in Album",
                    album.tracks_number as "Alsum Size"
                    from album inner join track
                    on album.id = track.album_id
                    where track.len <= \'{max_length}\'
                    and track.len >= \'{min_length}\';

        """

    elif "album size" == option:
        try:
            min_size, max_size = input(
                f"Enter minimal and maximum song number of songs in album,
separated by space\n\n").split()
            min_size, max_size = int(min_size), int(max_size)
        except Exception as _ex:
            raise ValueError("Invalid input. 2 parameters or <integer> type
were not provided. Try again.")

        sql = f"""select artist.name as "Group Name",
                    album.title as "Album Title",
                    album.tracks_number as "Album Size"
                    from artist inner join album
                    on album.artist_id = artist.id
                    where album.tracks_number <= {max_size}
                    and album.tracks_number >= {min_size};"""

    start = time()
    cursor.execute(sql)
    stop = time()
    logs = self.get_pretty_table(cursor.fetchall(),
tuple(array(cursor.description)[: , 0]))
    return logs, stop - start

```

Опис програмних функцій

В даних програмних модулях знаходяться лише функції, які реалізовані без hard-code. Представлений функціонал – працює не знаючи, які таблиці є в базі даних, які там атрибути та значення.

Завдання 2 й 3 були реалізовані в інших модулях, так як додавання їх сюди – порушувало б концепцію контролер-модель-подання семантично.

1. `connect_to_db()` - виконує з'єднання з базою даних
2. `disconnect_from_db()` - розриває з'єднання з базою даних
3. `connect()` - декоратор, котрий використовується в багатьох функціях, на випадок розриву з'єднання з базою даних з будь-яких причин.
4. `insert()` - виконує вставку даних до потрібної таблиці за одним або декількома атрибутами.
5. `select()` - отримує дані з бази даних з потрібної таблиці за одним, декількома або усіма атрибутами.
6. `update()` - оновляє дані у шуканій таблиці за одним або декількома атрибутами. Використовує один параметр як фільтрація рядків для оновлення.
7. `delete()` - видаляє дані з шуканої таблиці за значенням атрибуту, або просто видаляє все значення з таблиці.
8. `get_table_columns()` - повертає кортеж рядків, що є назвами стовпців для шуканої таблиці.
9. `class SQLGenerator(Analyser)` – клас `Analyser` використовується для більш складного процесу генерації. В рамках цієї лабораторної роботи, він не потрібен. Об'єкт класу `SQLGenerator` має поля `модель`, як посилання на об'єкт моделі, та два параметри `xml_source` & `is_empty`. Це потрібно для того, щоб створити таблиці з бібліотеки `itunes` у випадку, якщо база даних порожня.
`Self.text_request` – класичний приклад генерації текстового поля. Використовується для багатьох атрибутів, тому був реалізований як поле класу.
10. `create_tables` – створення бази даних з xml документу. Не входить в рамки лабораторної роботи. Було реалізовано за власним бажанням.
11. `get_if_unique()` - функція котра повертає `True` у випадку, якщо згенероване значення унікальне для значень цього атрибуту, інакше повертає `False`.
12. `generate_one_unique_attr_table()` - функція для генерації даних, таблиці, котра складається лише з ключового атрибуту та неключового.
13. `generate_table()` - загальна функція, котра генерує дані для усіх 4 таблиць.

14. `comprehensive_select()` - програма реалізація 3 завдання. З початку пропонує користувачу обрати режим роботи. Наразі підтримується лише режим `select`. Потім просить обрати опцію роботи в обраному режимі. У випадку некоректного вводу – йде валідація і повторна пропозиція обрати режим та опцію роботи.

Надалі йде реалізація пошуку за декількома атрибутами з декількох таблиць у випадку пошуку за діапазону років видання пісні, її тривалості або кількості пісень, що складають альбом. Функція повертає час виконання та текстове відображення шуканих рядків.