

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**»**

Виконав студент ІII курсу

ФПМ групи КВ-83

Мельник Ю.Ю.

Перевірив: Павловський В.І.

Київ – 2020

**Завдання**

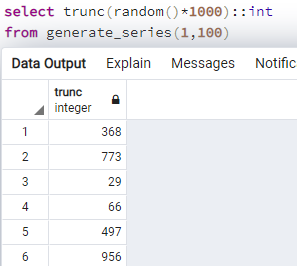
*Загальне завдання* роботи полягає у наступному:

1. Реалізувати функції 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).

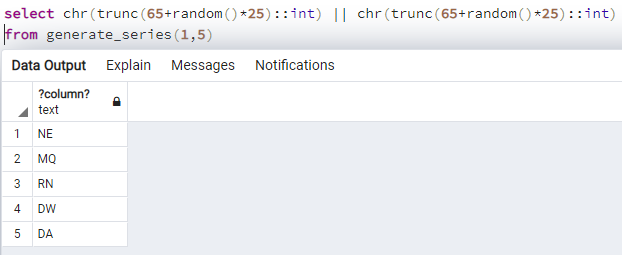
*Деталізоване завдання*:

1. Забезпечити можливість уведення/редагування/вилучення даних у таблицях бази даних з можливістю контролю відповідності типів даних атрибутів таблиць (рядків, чисел, дати/часу). Для контролю пропонується два варіанти: контроль при введенні (валідація даних) та перехоплення помилок (try..except) від сервера PostgreSQL при виконанні відповідної команди SQL. Особливу увагу варто звернути на дані таблиць, що мають зв’язок 1:N. При цьому з боку батьківської таблиці необхідно контролювати **вилучення** рядків за умови наявності даних у підлеглій таблиці. З точки зору підлеглої таблиці варто контролювати наявність відповідного рядка у батьківській таблиці при виконанні **внесення** нових даних. Унеможливити виведення програмою системних помилок на екрані шляхом їх перехоплення і адекватної обробки. Внесення даних виконується користувачем у консольному вікні програми.
2. Забезпечити можливість автоматичної генерації великої кількості даних у таблицях за допомогою вбудованих у PostgreSQL функцій роботи з псевдовипадковими числами. Дані мають бути згенерованими **не мовою програмування, а відповідним SQL-запитом**!

*Приклад генерації 100 псевдовипадкових чисел:*



*Приклад генерації 5 псевдовипадкових рядків:*



Приклад генерації псевдовипадкової мітки часу з діапазону [доступний за посиланням](https://stackoverflow.com/questions/22964272/postgresql-get-a-random-datetime-timestamp-between-two-datetime-timestamp/22965061).

Кількість даних для генерування має вводити користувач з клавіатури. Для тесту взяти 100 000 записів для однієї-двох таблиць.

Особливу увагу слід звернути на відповідність даних вимогам зовнішніх ключів з метою уникнення помилок порушення обмежень цілісності (foreign key).

1. Для реалізації пошуку необхідно підготувати 3 запити, що включають дані з декількох таблиць і фільтрують рядки за 3-4 атрибутами цих таблиць. Забезпечити можливість уведення конкретних значень констант для фільтрації з клавіатури користувачем. Крім того, після виведення даних необхідно вивести час виконання запиту у мілісекундах. Перевірити швидкодію роботи запитів на попередньо згенерованих даних.
2. Програмний код організувати згідно шаблону Model-View-Controller(MVC).  Приклад організації коду згідно шаблону доступний [за даним посиланням](https://www.giacomodebidda.com/mvc-pattern-in-python-introduction-and-basicmodel/). При цьому модель, подання та контролер мають бути реалізовані у окремих файлах. Для доступу до бази даних використовувати **лише мову SQL** (без ORM).

Рекомендована бібліотека взаємодії з PostgreSQL Psycopg2: <http://initd.org/psycopg/docs/usage.html>)

**Нормалізована модель бази даних**

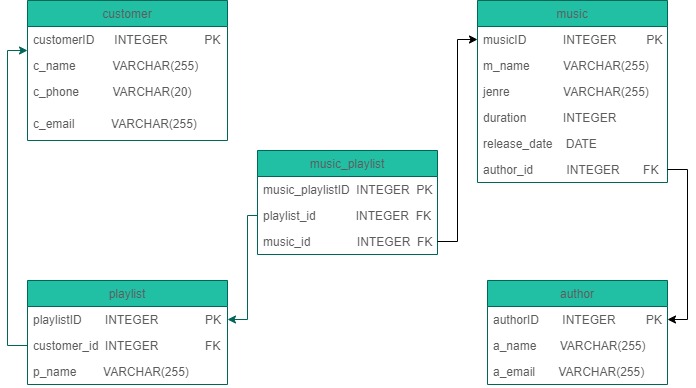


Рис 2.1 – Нормалізована модель даних

**Структура програми**

Програма створена за патерном MVC (Model-View-Controller). Складається відповідно з модулів Model , View та Controller.

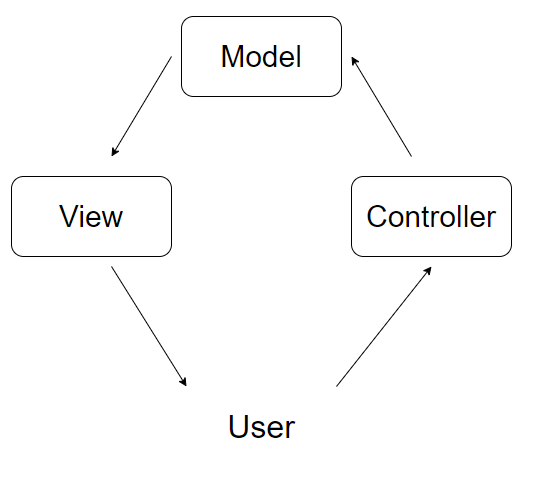


Рис 2.2 – Модель MVC

**Опис програми**

У класі Model реалізовані функції , що здійснюють SQL запити до Бази Даних.

У класі View реалізовані функції, що використовуються для відображення в консоль пунктів меню та виводу даних з таблиць.

У класі Controller реалізовані функції для відповідних меню та допоміжні функції.

В модулі db\_backend знаходиться функція, що виконує з’єднання з БД та функції для додавання, видалення, та оновлення записів.

**Опис структури меню програми**

1. Пункт ‘Редагувати дані’ складається з підпунктів включає в себе базові операції з таблицями БД:

* Додати запис
* Оновити дані запису
* Відобразити дані
* Видалити дані
* Вихід

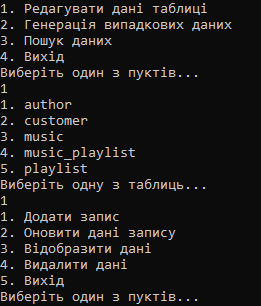


Рис 2.3 – Пункт ‘Редагувати дані’

1. Пункт ‘Генерація випадкових даних’ дозволяє згенерувати введену кількість полів в вибраній таблиці.
2. Пункт ‘Пошук даних’ проводить пошук, який вибрав користувач.

Складається з підпунктів:

* Пошук пісні, яка вийшла після певної дати та певного автора;
* Пісня певною тривалістю, її автор та плейлист в якому вона знаходиться;
* Пісні, реліз яких відбувся після певної дати та які мають певні словосполучення;

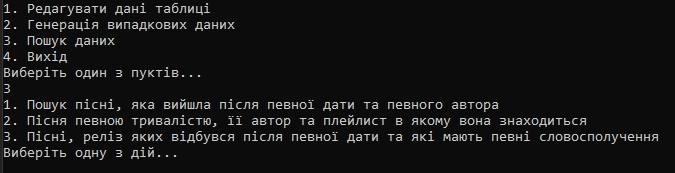


Рис 2.4 – Пункт ‘Пошук даних’

1. Пункт ‘Вихід’ вихід з програми

**Середовище розробки**

Середовище для лагодження SQL-запитів до бази даних – PgAdmin4.

Мова програмування – Python 3.8.5

Бібліотека взаємодії з PostgreSQL Psycopg2.

Текстовий редактор – Sublime Text.

**Вибірка елементів з БД**

Вибірка елементів з таблиці “author”

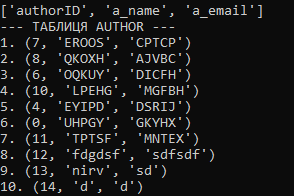


Рис 2.5 – Елементи таблиці author

Вибірка елементів з таблиці “customer”

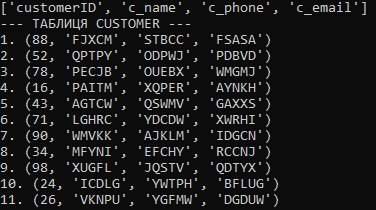


Рис 2.6 – Елементи таблиці users

Метод, який дає змогу отримати дані таблиці:

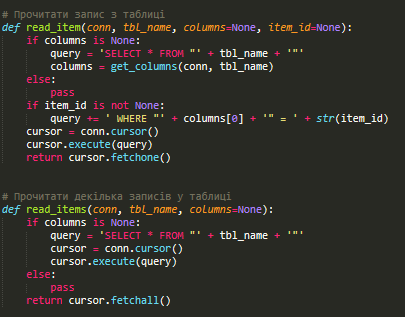


Рис 2.7 – Методи для отримання даних з таблиці

**Видалення даних з таблиці**

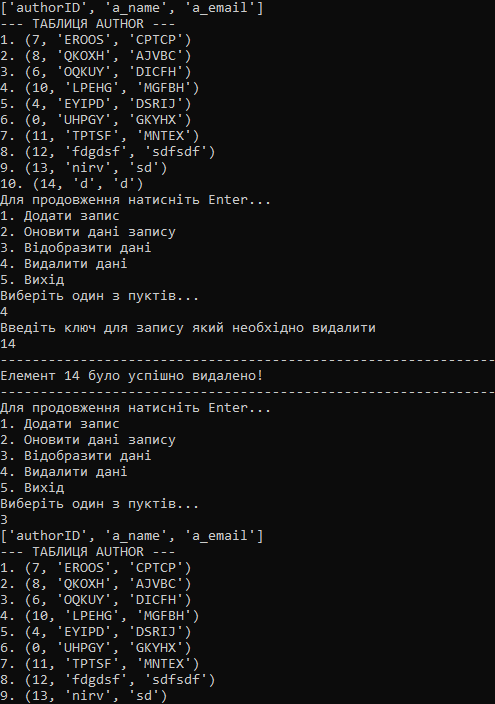


Рис 2.8 – Приклад видалення з таблиці author

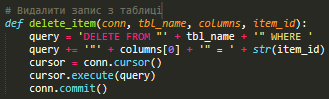


Рис 2.9 – Метод видалення даних з таблиці

**Додавання даних до таблиці**

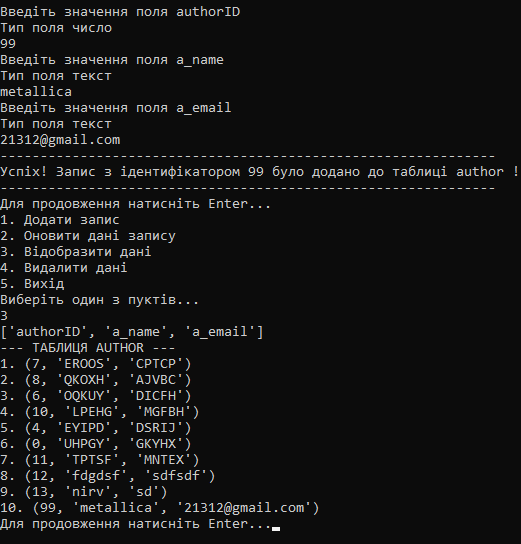


Рис 2.10 – Результат додавання

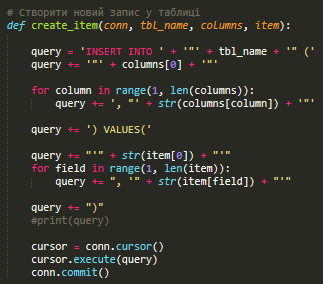


Рис 2.11 – Метод додавання даних до таблиці

**Зміна запису в таблиці**

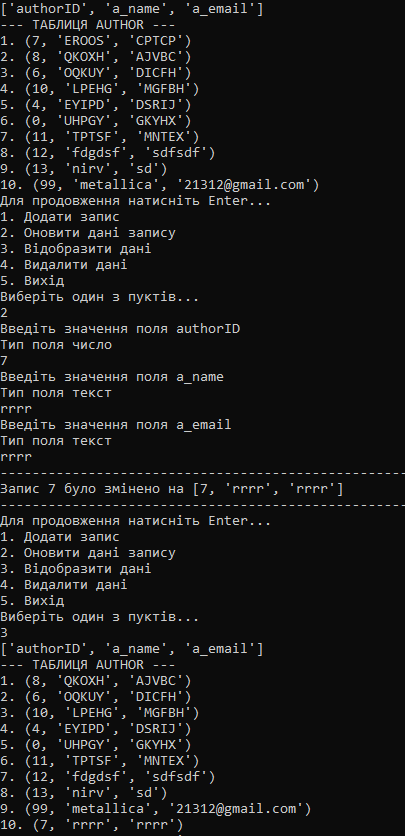


Рис 2.12 – Результат зміни даних для таблиці author

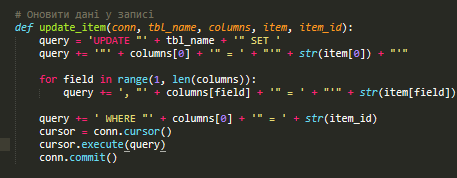


Рис 2.13 – Метод для оновлення даних

**Генерування N рядків до таблиці**

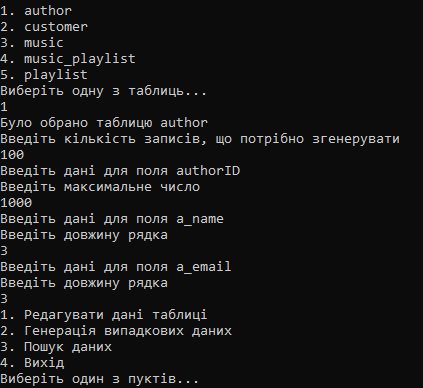


Рис 2.14 – Меню генерування

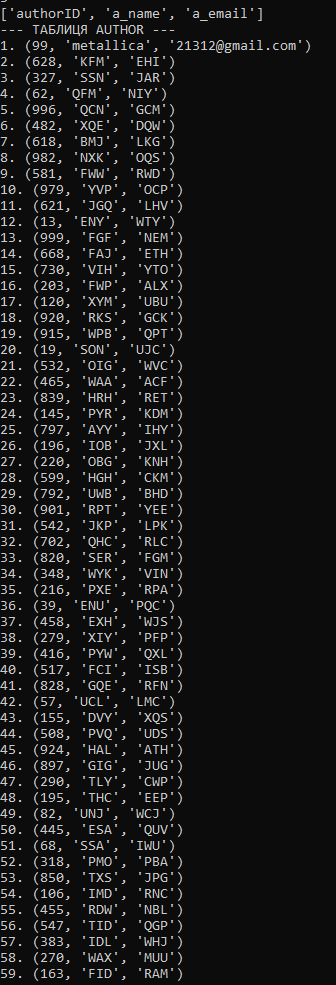


Рис 2.15 – Результат генерування рядків до таблиці author



Рис 2.16 – Метод генерування рядків.

**Видалення записів з батьківської таблиці**

При видаленні запису з батьківської таблиці автоматично будуть видалені записи з дочірньої таблиці, які відповідали видаленим даним батьківської таблиці.

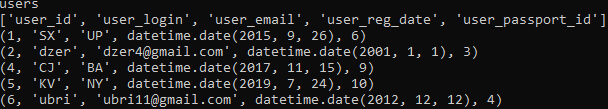
****

Рис 2.17 – Дочірня таблиця до видалення запису

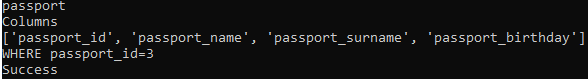
****

Рис 2.18 – Видалення запису з батьківської таблиці

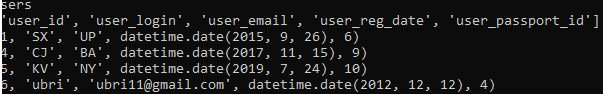
****

Рис 2.19 – Дочірня таблиця після видалення запису

**Видалення записів з батьківської таблиці**

В програмі реалізовано перехоплення помилок з сервера SQL. Тобто при спробі видалення з батьківскої таблиці рядок, який св’язаний з рядком з дочірньої таблиці отримаємо помилку про порушення обмеження зовнішнього ключа  
.

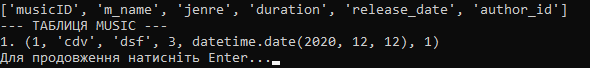


Рис 2.20 – Дочірня таблиця

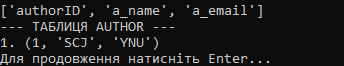


Рис 2.21 – Батьківська таблиця

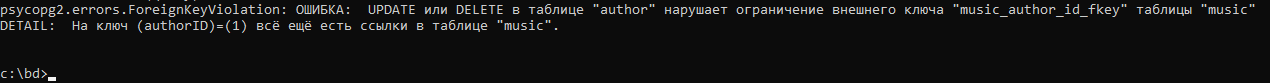


Рис 2.22 – Результат спроби видалення рядка з батьківської таблиці

**Операція додавання рядку в дочірню таблицю**

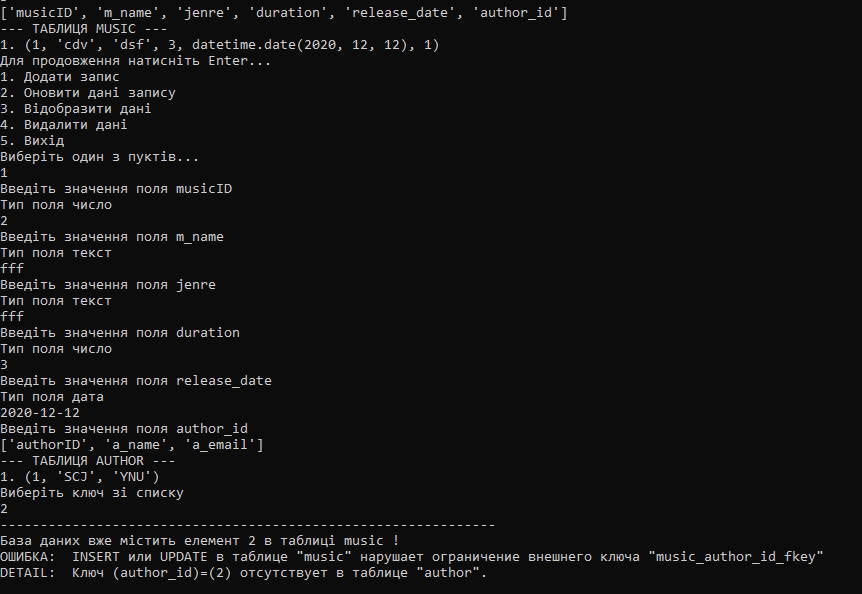


Рис 2.23 – Помилка при вставці запису в дочірню таблицю

При спробі додавання в дочірню таблицю елемента отримуємо повідомлення про помилку, оскільки в батьківській таблиці відсутні дані до яких користувач хоче додати запис.

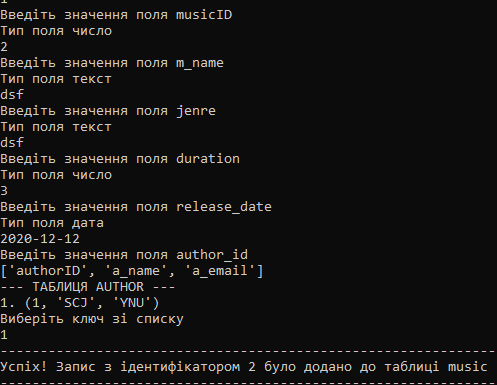


Рис 2.24 – Спроба вставки даних в дочірню таблицю

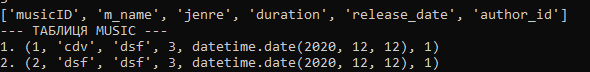


Рис 2.25 – Результат вдалої вставки в дочірню таблицю

**Пошук в таблицях**

У програмі реалізовано три запиту пошуку:

* 1. Пошук пісні, яка вийшла після певної дати та певного автора.
  2. Пошук пісні певною тривалістю, її автор та плейлист в якому вона знаходиться.
  3. Пошук пісні по словополученню з назви та певного жанру.

Перший пошук використовує дані з двох таблиць, другий з чотирьох таблиць, а третій запит виконує пошук по таблиці music.

**Перший пошук:**



Рис 2.26 – Таблиця author

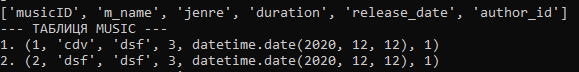


Рис 2.27 – Таблиця music

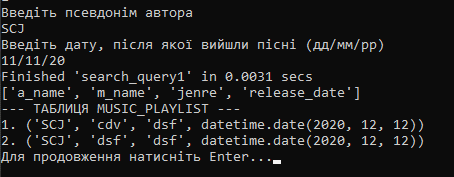


Рис 2.28 – Результат першого пошуку

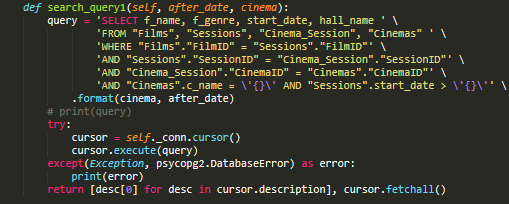


Рис 2.29 – метод першого пошуку

**Другий пошук:**

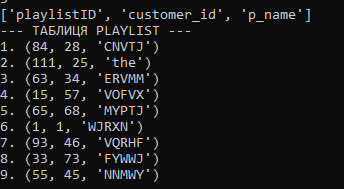


Рис 2.30 – Таблиця playlist

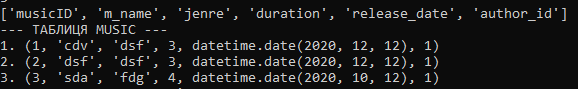


Рис 2.31 – Таблиця music

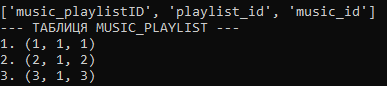


Рис 2.32 – Таблиця music\_playlist

C:\Users\sobak\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Снимок экрана (391).png

Рис 2.33 – Таблиця author

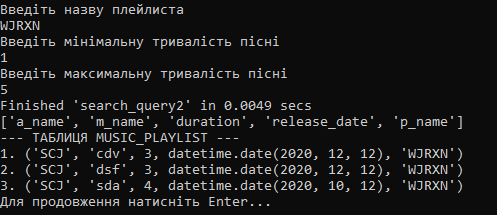


Рис 2.34 – Результат другого пошуку

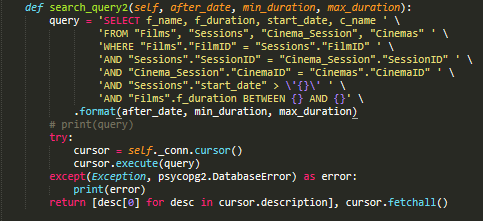


Рис 2.35 – Метод другого пошуку

**Третій пошук:**

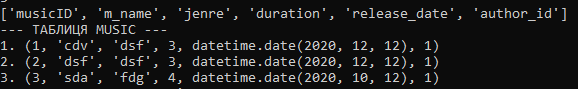


Рис 2.36 – Таблиця music

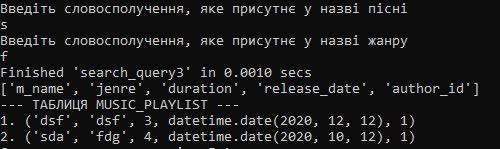


Рис 2.37 – Результат третього пошуку

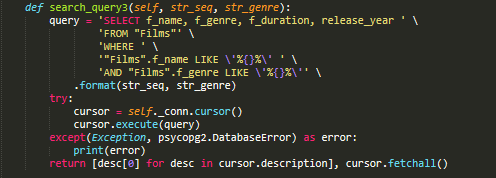


Рис 2.38 – Метод третього пошуку

**Модуль “model”**



Рис 2.39 – Модуль “model ”

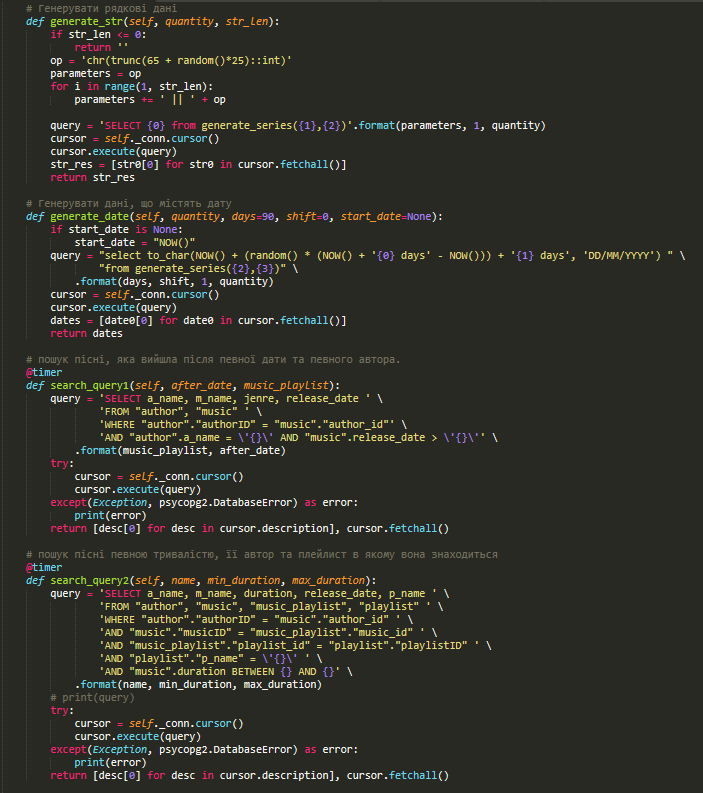


Рис 2.40 – Модуль “ model ”



Рис 2.41 – Модуль “ model ”

**Модуль “view”**



Рис 2.42 – Модуль “view”

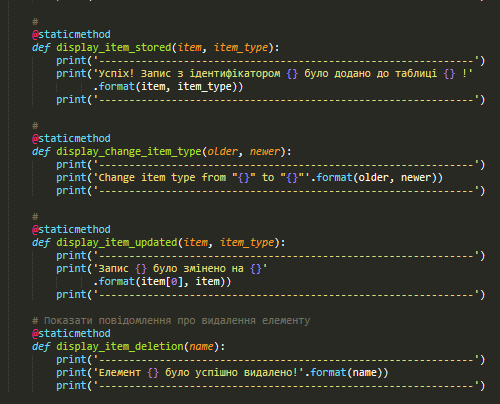


Рис 2.43 – Модуль “view”

**Модуль “controller”**

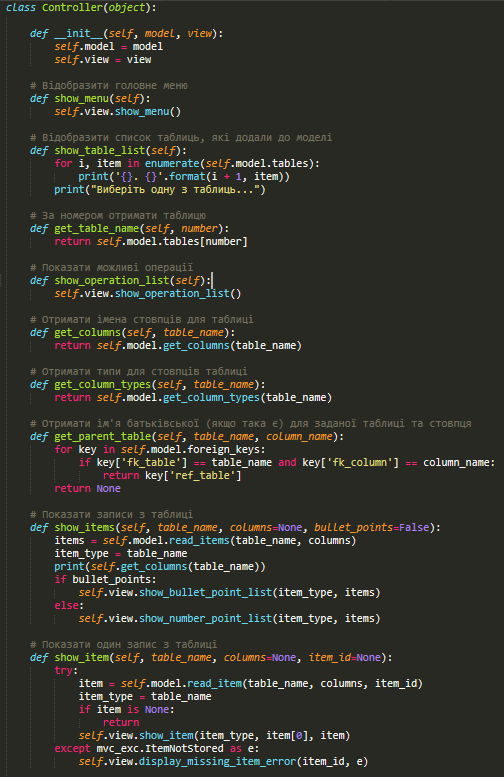


Рис 2.44 – Модуль “controller”

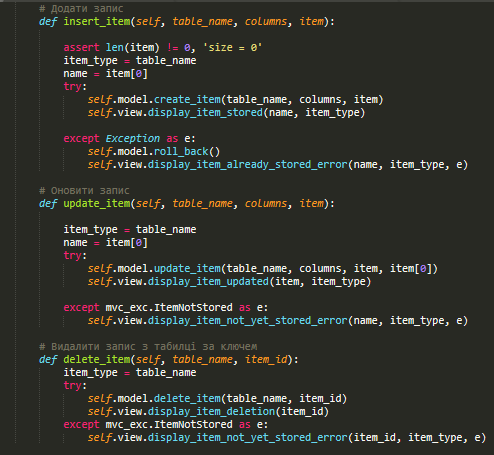


Рис 2.45 – Модуль “controller”

**Модуль “db\_backend”**



Рис 2.46 – Модуль “db\_backend”

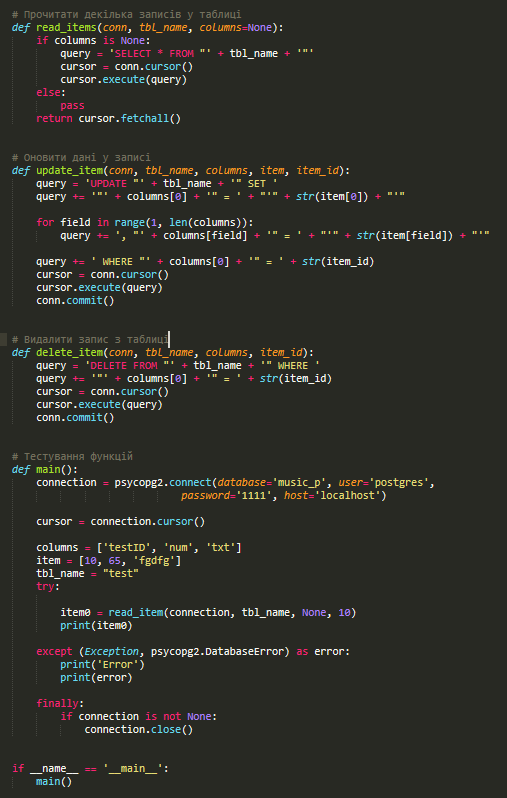


Рис 2.47 – Модуль “db\_backend”

**Модуль “main”**



Рис 2.48 – Модуль “main”



Рис 2.49 – Модуль “main”



Рис 2.50 – Модуль “main”