

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

Панфілов Я.Ю.

Перевірив: Павловський В.І.

Київ – 2020

**Ознайомлення з базовими операціями СУБД PostgreSQL**

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

*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 2-х 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.

Програмний код виконати згідно шаблону MVC (модель-подання-контролер).

**Нормалізована модель бази даних**

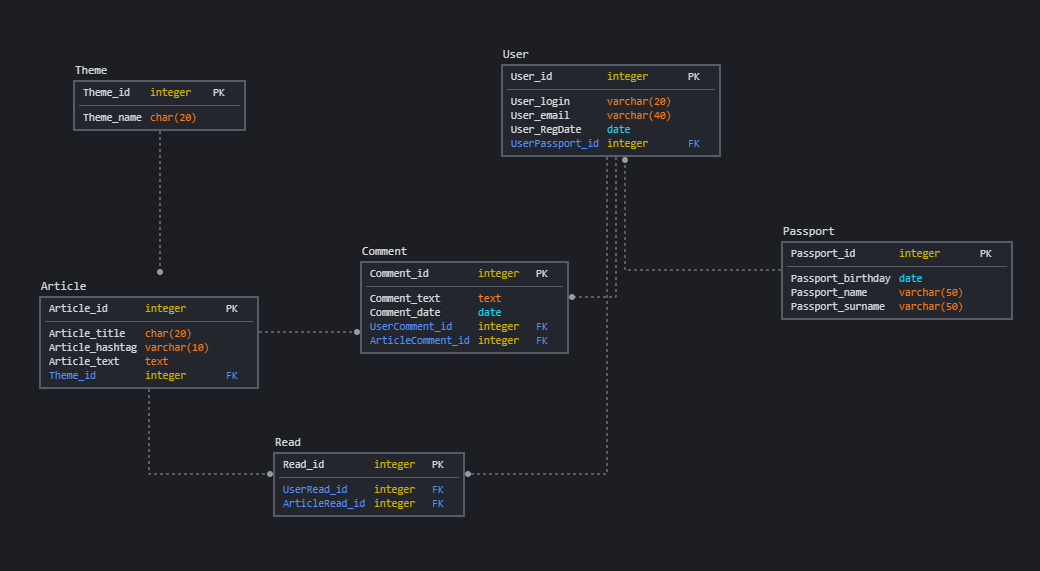


Рис 2.1 – Нормалізована модель даних

**Структура програми**

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

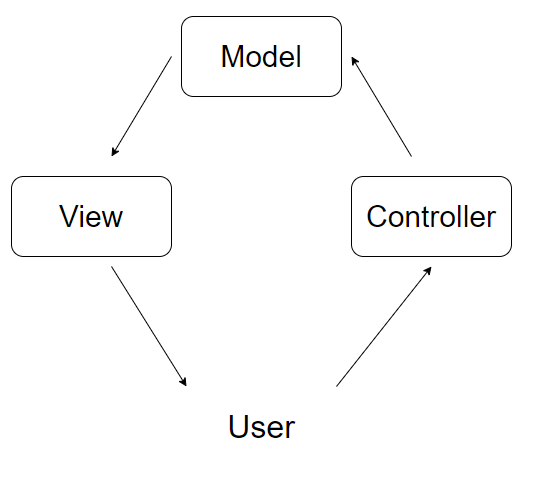


Рис 2.2 – Модель MVC

**Опис програми**

У класі db\_model реалізовані функції , що здійснюють SQL запити до Бази Даних, а також функція, що виконує з’єднання з БД.

У класі console\_view реалізовані функції, що використовуються для відображення в консоль пунктів меню, виводу даних з таблиць, тобто функції, що відображують певну інформацію в консоль.

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

**Опис структури меню програми**



Рис 2.3 – Меню програми

**Інтерфейс програми**

1. Пункт Show виводить дані про вміст вибраної таблиці.
2. Пункт Insert вставляє введені дані в вибрану таблицю. При введені некоректних даних програма покаже помилку.
3. Пункт Delete дає змогу видалити дані з вибраної таблиці за введеною умовою. При введені неправильної умови програма покаже помилку.
4. Пункт Change дозволяє змінити дані в вибраній таблиці. При введені некоректних даних програма покаже помилку.
5. Пункт Generate дозволяє згенерувати введену кількість полів в вибраній таблиці.
6. Пункт Query проводить пошук в вибраній комбінації таблиці та за певною умовою. При введені неправильних даних програма покаже помилку.
7. Пункт Quit виконує функцію виходу з програми.

**Середовище розробки**

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

Мова програмування – Python 3.7.

Бібліотека взаємодії з PostgreSQL Psycopg2.

Середовище розробки програмного забезпечення – Visual Studio 2019 та Visual Studio Code.

**Вибірка елементів з БД**

Вибірка елементів з таблиці “article”

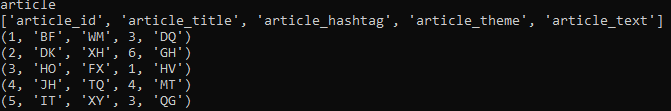
****

Рис 2.4 – Елементи таблиці article

Вибірка елементів з таблиці “users”

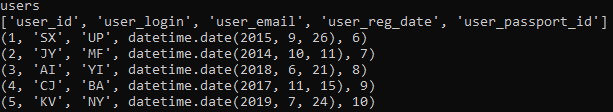


Рис 2.5 – Елементи таблиці users

Метод, який дає змогу отримати дані таблиці

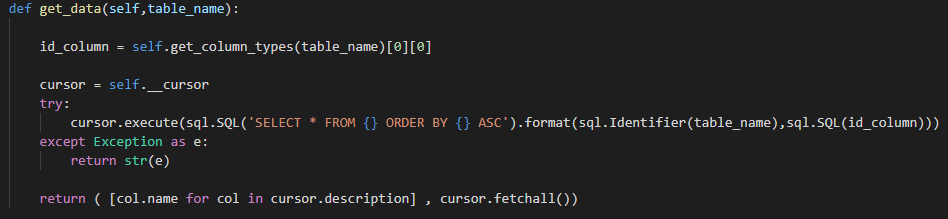


Рис 2.6 – Метод для отримання даних з таблиці

**Видаленні даних з таблиці**

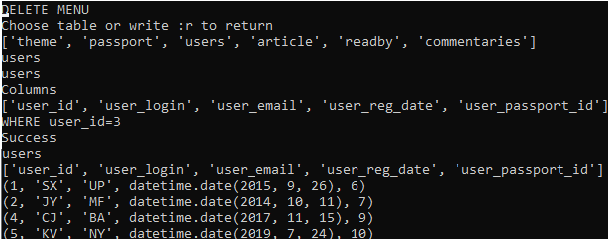


Рис 2.7 – Приклад видалення з таблиці users

При видаленні даних з однієї таблиці будуть видалені всі пов’язані записи з інших таблиць, що пов’язані з першою.

Метод, який видаляє дані з таблиці

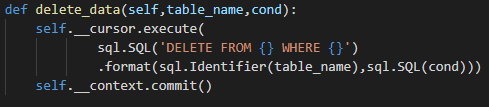


Рис 2.8 – Метод видалення даних з таблиці

**Додавання даних до таблиці**

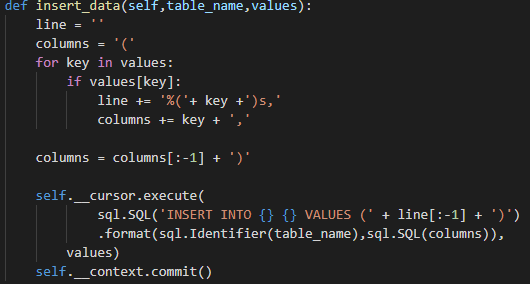
****

Рис 2.9 – Метод додавання даних до таблиці

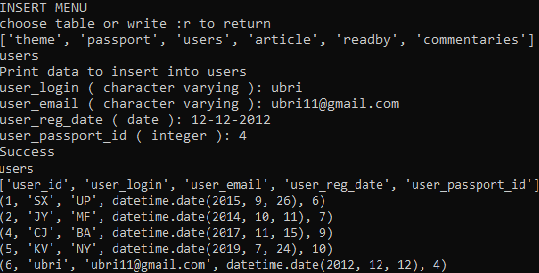


Рис 2.10 – Результат додавання запису

**Зміна запису в таблиці**

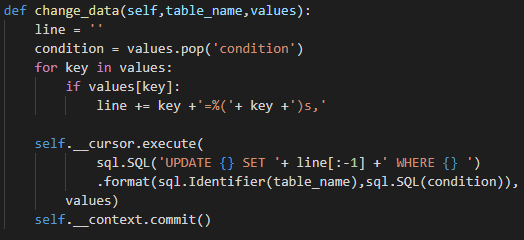
****

Рис 2.11 – Метод для зміни даних в таблиці

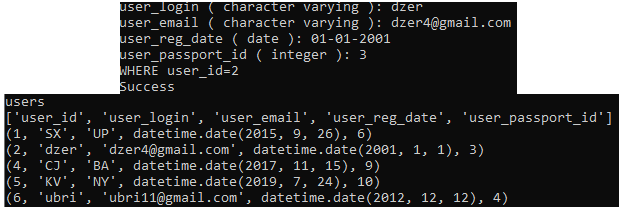


Рис 2.12 – Результат зміни даних для таблиці users

**Пошук в декількох таблицях**

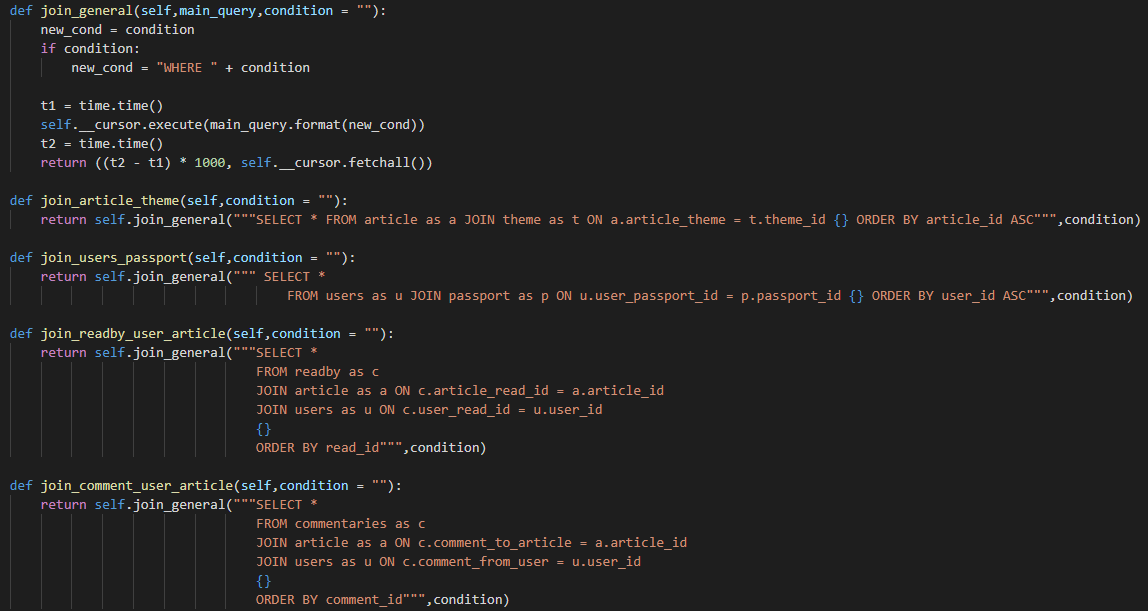
****

Рис 2.13 – Методи для пошуку даних в декількох таблицях

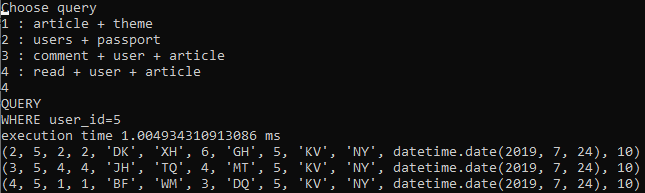


Рис 2.14 – Результат пошуку для таблиць readby, users, article

**Додавання N рядків до таблиці**

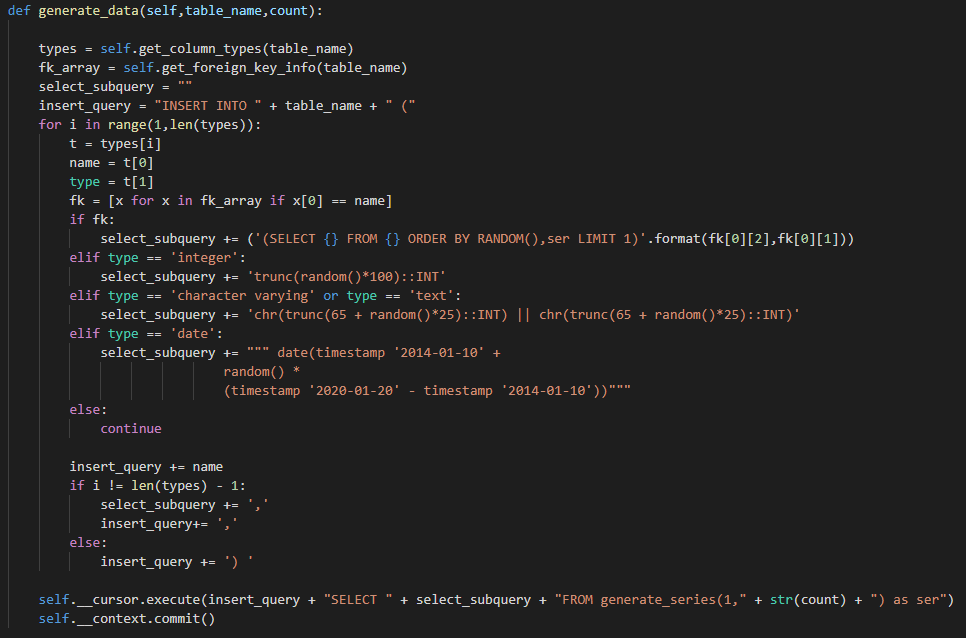
****

Рис 2.15 – Метод для генерування даних до таблиці

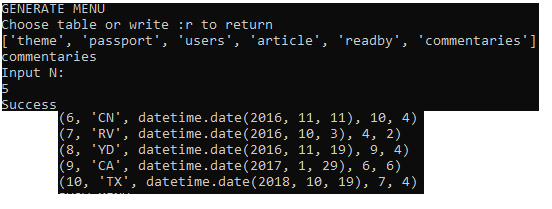


Рис 2.16 – Результат додавання 5 рядків до таблиці commentaries

**Лістинг модуля “Model”**

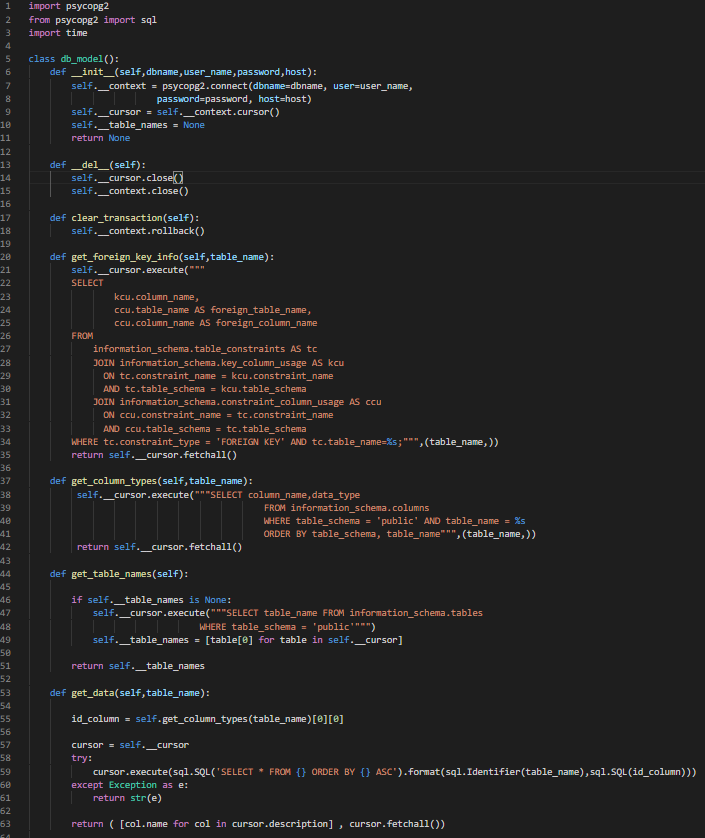
****

Рис 2.17 – Модуль “Model”

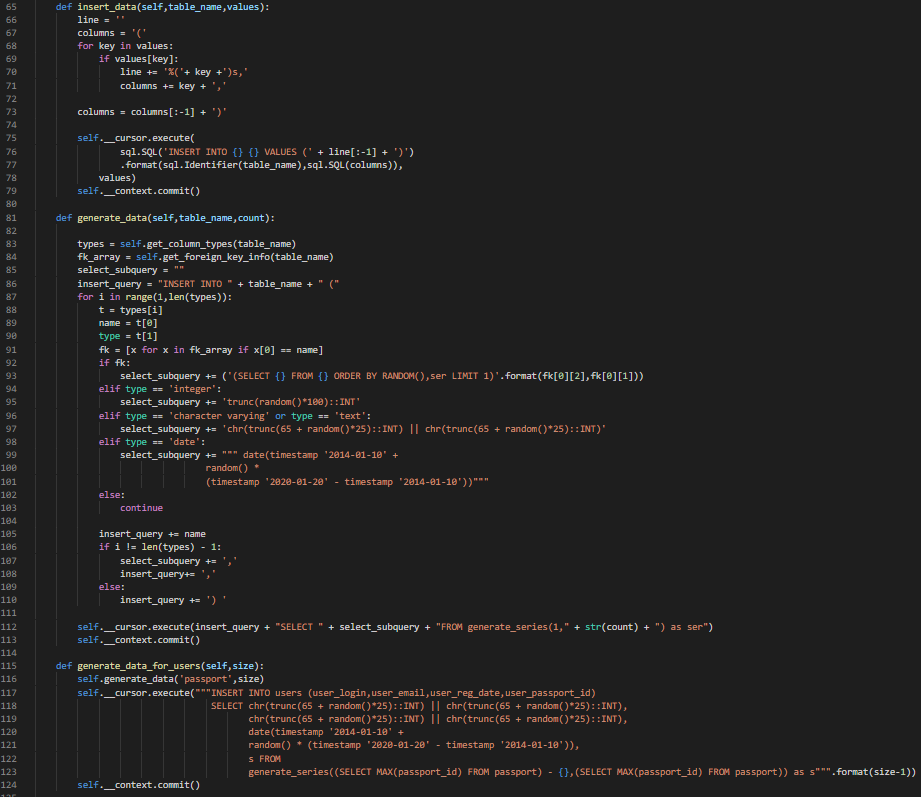
****

Рис 2.18 – Модуль “Model”

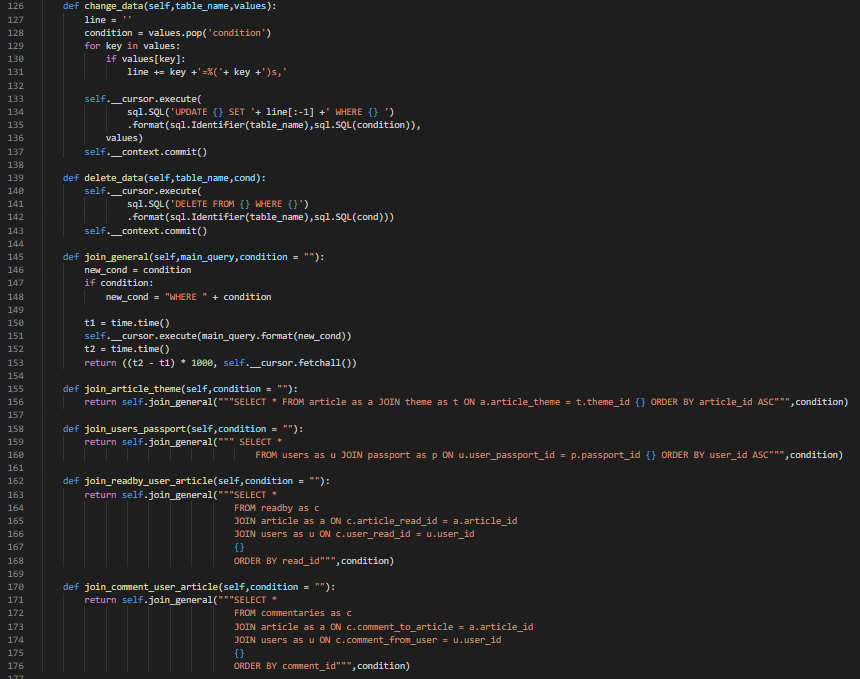


Рис 2.19 – Модуль “Model”

**Видалення записів з батьківської таблиці**

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

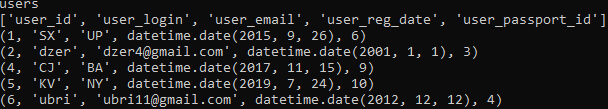
****

Рис 2.20 – Дочірня таблиця до видалення запису

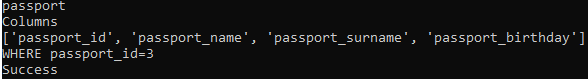
****

Рис 2.21 – Видалення запису з батьківської таблиці

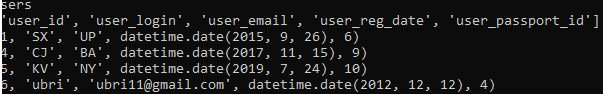
****

Рис 2.22 – Дочірня таблиця після видалення запису

**Операція вставки запису в дочірню таблицю**

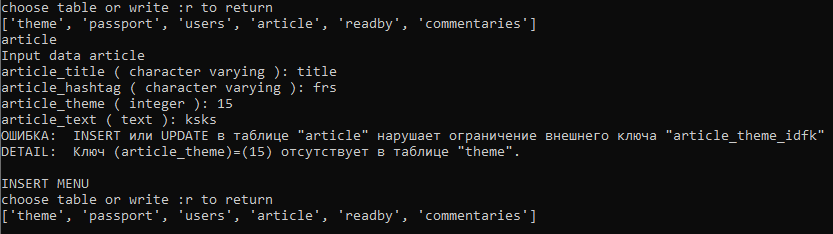
****

Рис 2.23 – Помилка при вставці запису в дочірню таблицю

При спробі вставці запису в дочірню таблицю отримано повідомлення про помилку, оскільки в батьківській таблиці відсутні дані до яких користувач хоче додати запис. Після повідомлення помилки видано меню для повторного введення даних.

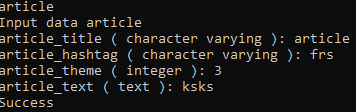
****

Рис 2.24 – Спроба вставки даних в дочірню таблицю

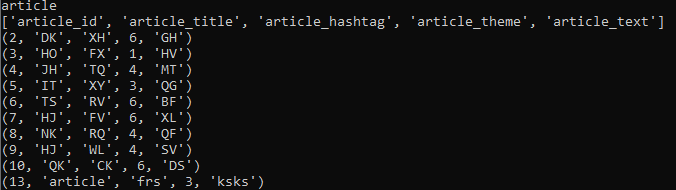
****

Рис 2.25 – Результат вдалої вставки в дочірню таблицю

**Модуль “Controller”**

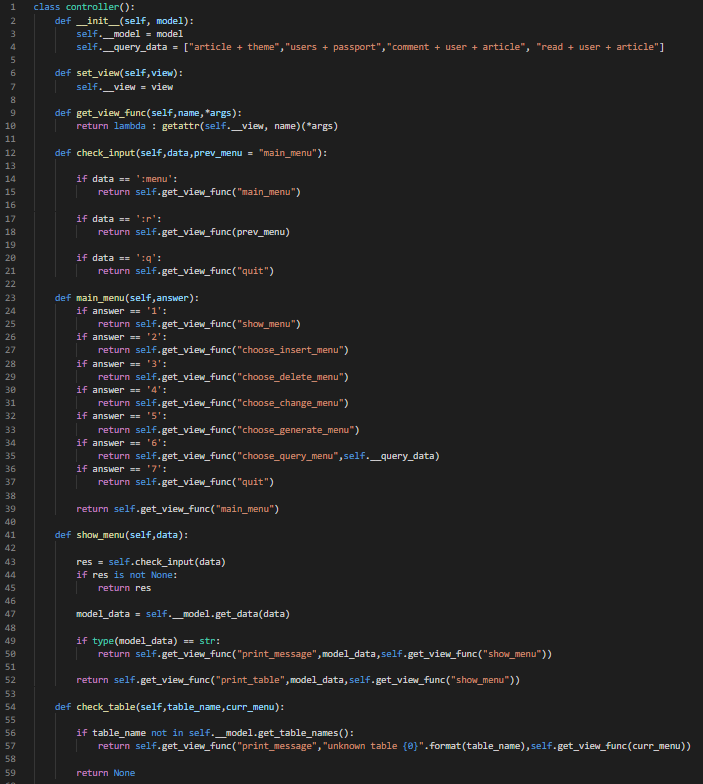
****

Рис 2.26 – Модуль “controller”

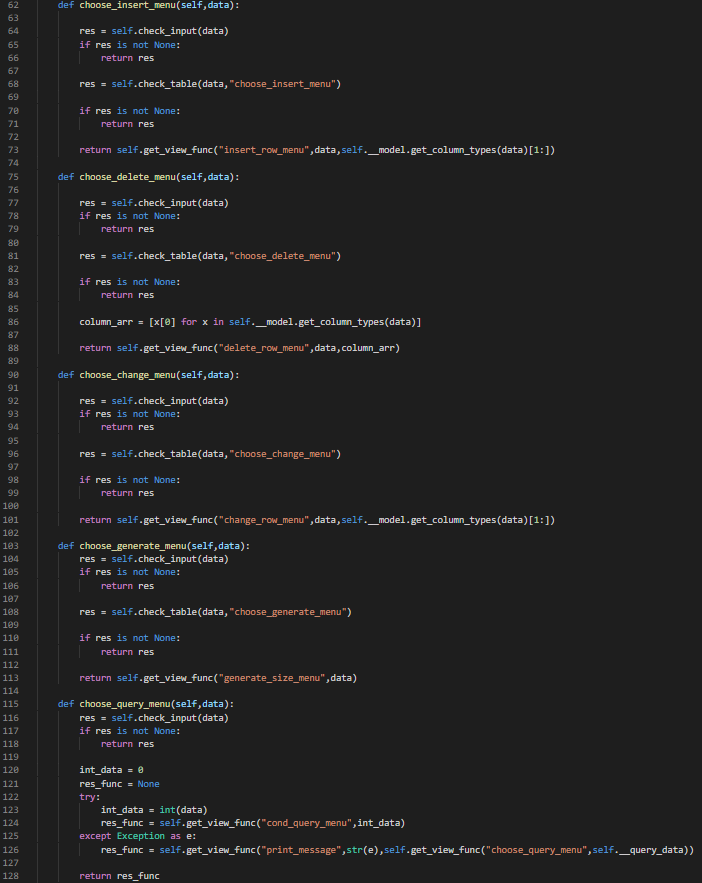
****

Рис 2.27 – Модуль “controller”

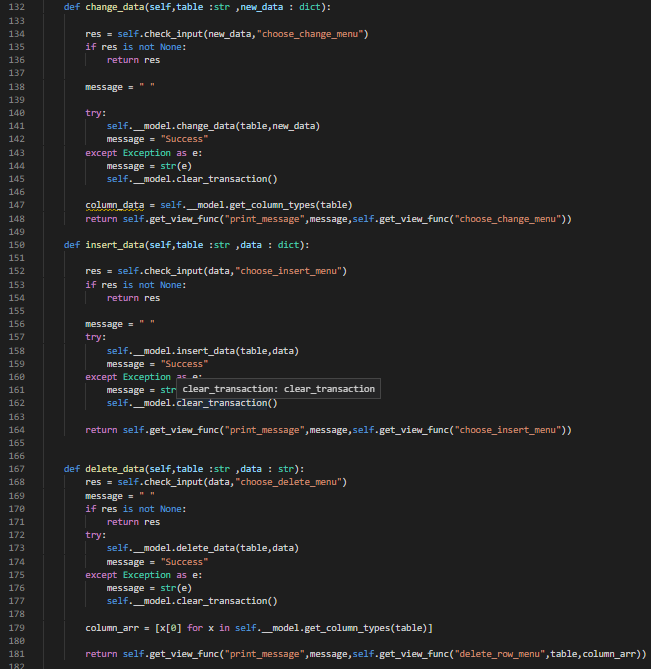
****

Рис 2.28 – Модуль “controller”

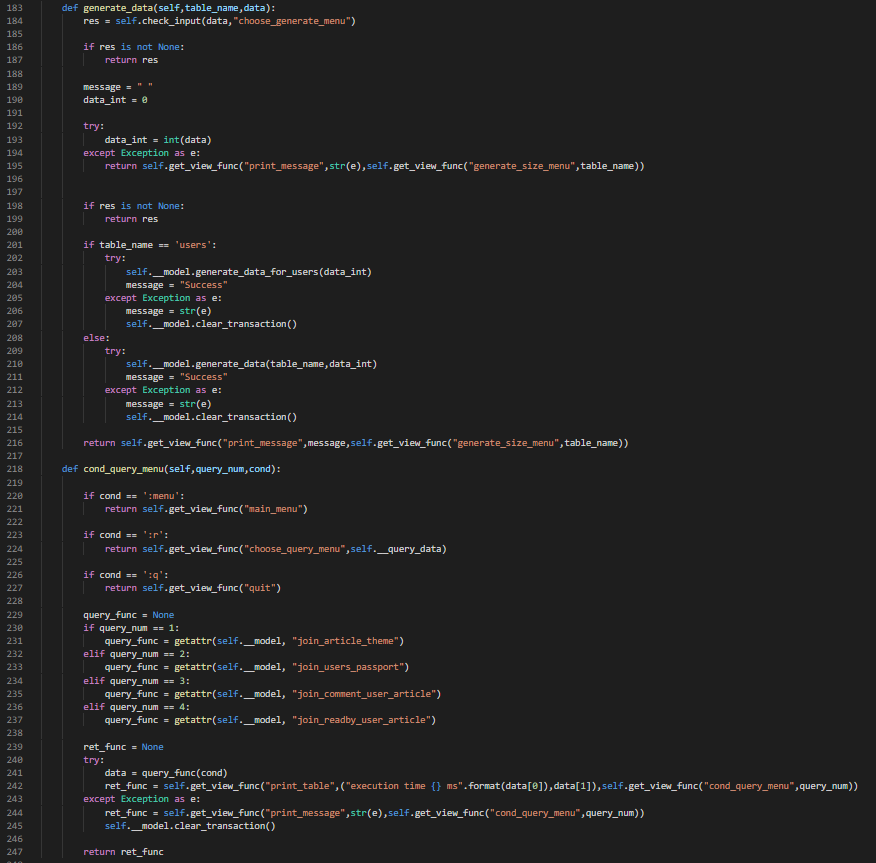
****

Рис 2.29 – Модуль “controller”

**Модуль “Console\_view”**

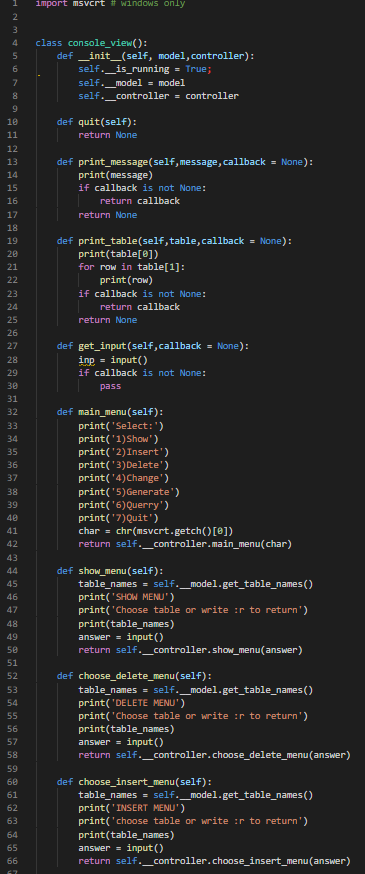


Рис 2.30 – Модуль “console\_view”



Рис 2.31 – Модуль “console\_view”