

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

Пащенко Антон КВ-83

Перевірив: Павловський В.І.

Київ – 2020

**Ознайомлення з базовими операціями СУБД PostgreSQL**

*Завдання* роботи полягає у наступному:

1. Виконати нормалізацію бази даних, яка була створена у лабораторній роботі №1, до третьої нормальної форми (3НФ);
2. Реалізувати функціональні вимоги, наведені нижче.

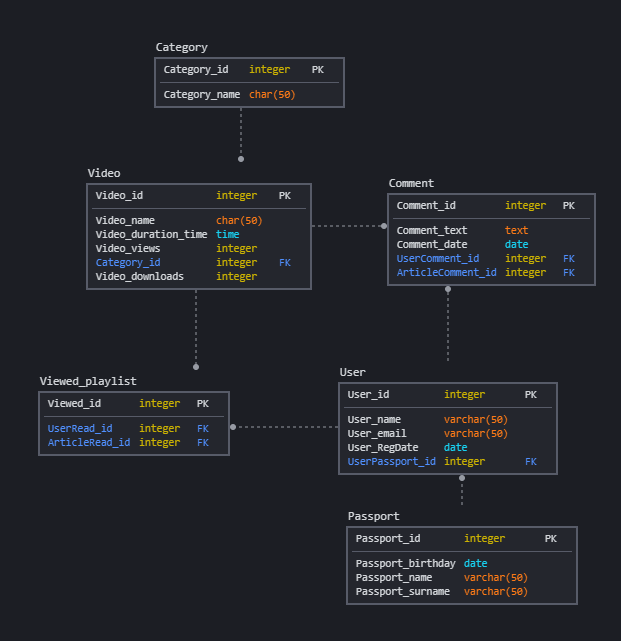
*Функціональні вимоги*

1. Реалізувати внесення, редагування та вилучення даних у базі засобами консольного інтерфейсу;
2. Передбачити автоматичне пакетне генерування «рандомізованих» даних у базі;
3. Забезпечити реалізацію пошуку за двома-трьома атрибутами з двох сутностей одночасно: для числових атрибутів – у рамках діапазону, для рядкових – як перелічення, для логічного типу – значення True/False, для дат – у рамках діапазону дат;
4. Забезпечити реалізацію повнотекстового пошуку за будь-яким текстовим атрибутом бази даних засобами PostgreSQL з виділенням знайденого фрагменту.

*Вимоги до інтерфейсу користувача*

1. Використовувати консольний інтерфейс користувача.

**Нормалізована логічна модель даних БД «Відеохостінг»**



**Рис 2.1 Нормалізована логічна модель даних БД «Відеохостінг»**

Усі таблиці (відношення) знаходяться в 3 НФ тому, що у кожній із них:

1. Всі атрибути є атомарними та відсутні повторення рядків (1НФ);
2. Первинний ключ складається лише з одного атрибуту (2НФ);
3. Кожний не первинний атрибут не є транзитивно залежним від первинного ключа (3НФ).

**Опис програми**

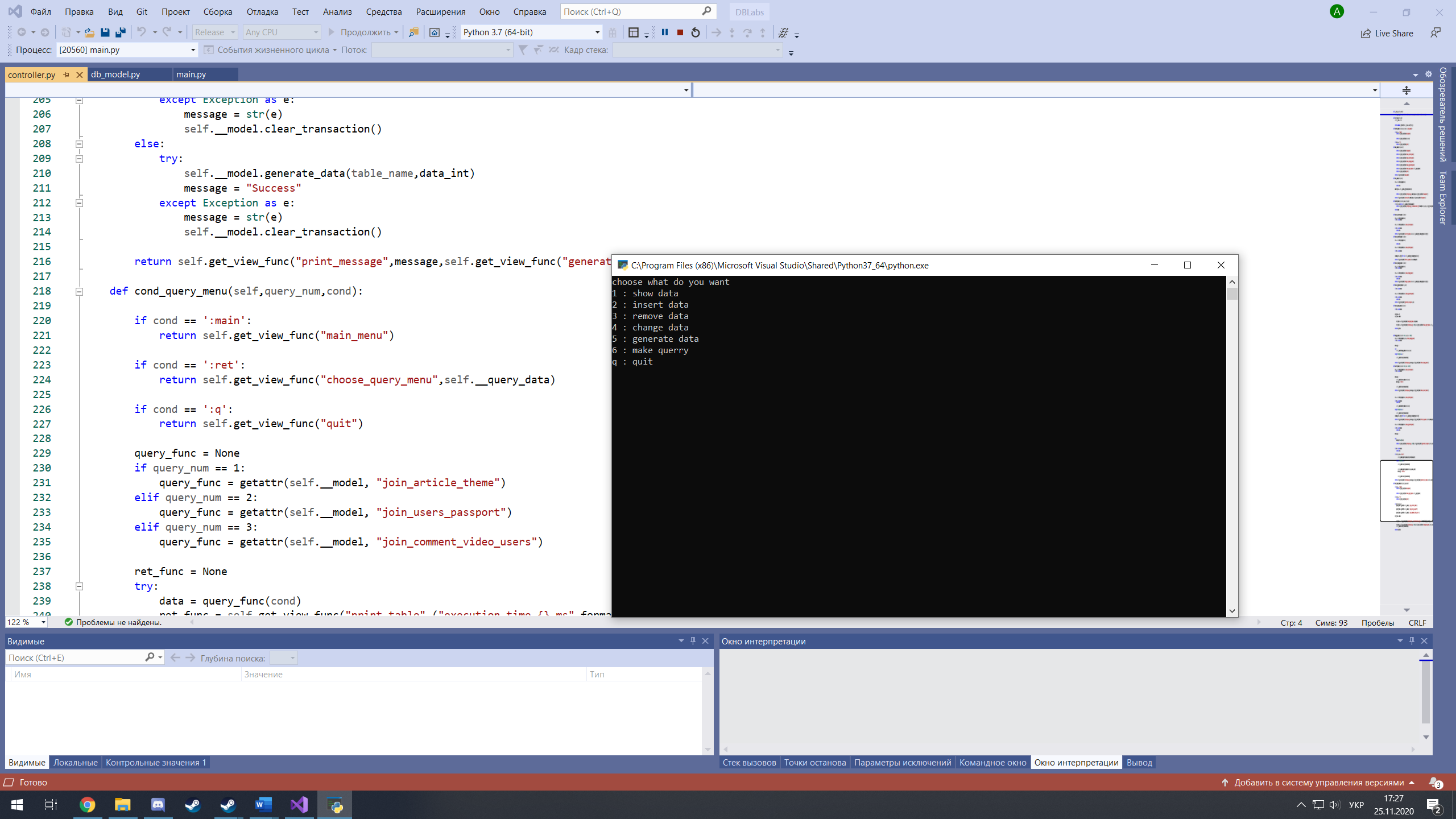
Програма створена для управління базою даних за допомогою базових операцій СУБД PostgreSQL та реалізовує функціональні вимоги, що наведені у завданні. Програма складається з 3 модулів:

1. console\_view – клас що відповідає за відображення данних та обробку вводу користувача.
2. controller.py – клас що обробляє запити від console\_view та викликає методи моделі.
3. db\_model – клас що виконує запити до PostgresSQL та повертає результат.

Використані сторонні бібліотеки psycopg2, prettytable.

**Структура меню програми**

Меню програми



**Рис 2.2** **Меню програми**

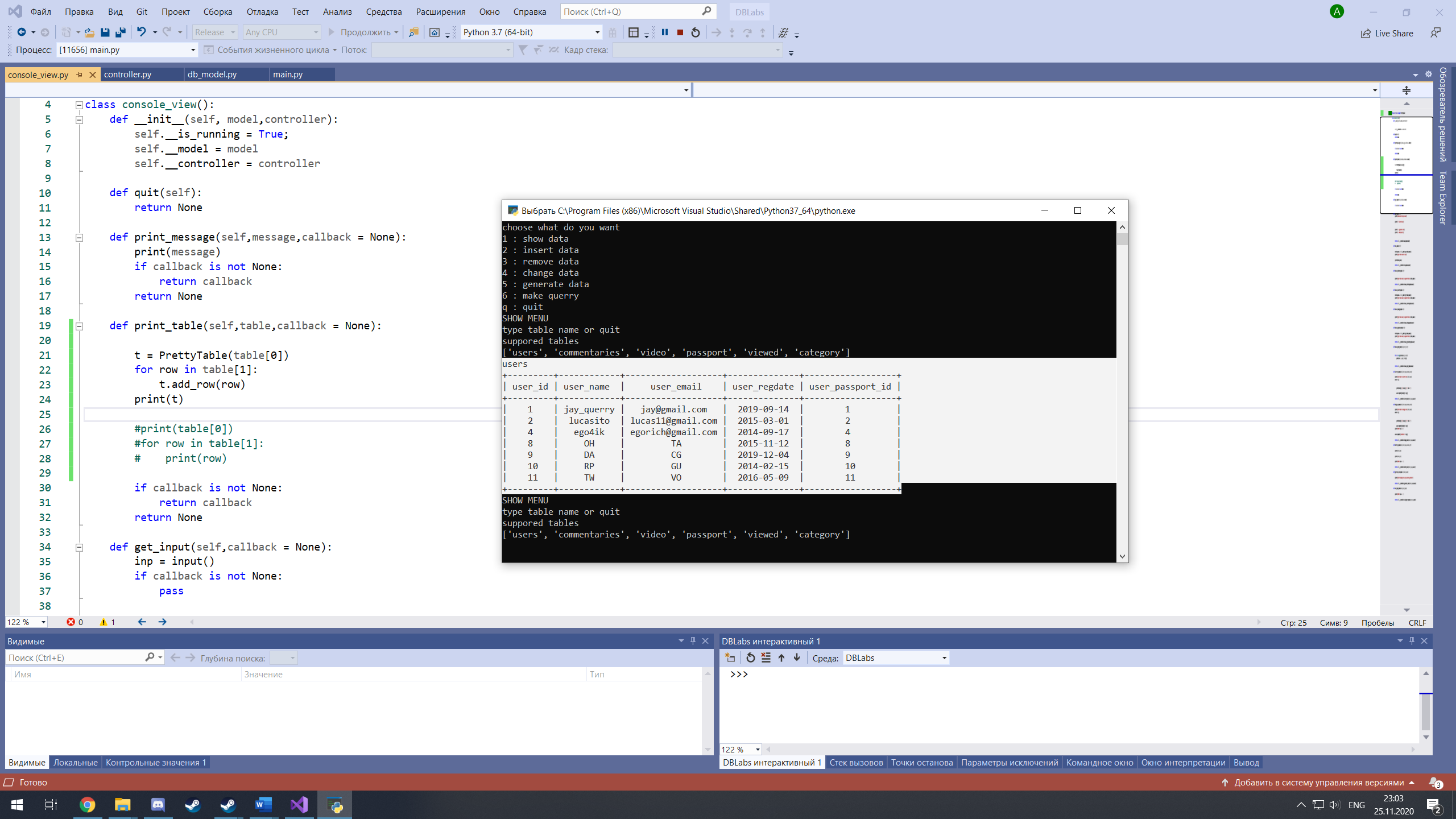
**Функціонал**

Кожне меню запитує таблицю з якою буде проведена якась дія та додаткову інформацію в залежності від дії. При введенні неіснуючої таблиці програма повідомить про помилку та дасть можливість ввести коректні данні.

1. Show data виводить стовпці таблиці.
2. Insert data запрошує данні для конкретної таблиці, при введенні некоректних даних (за типом) програма повідомить про помилку та дасть можливість ввести коректні данні. Після успішного вводу кортеж додається до таблиці.
3. Remove data запрошує умову за якою будуть видалятися дані, при введенні некоректної умови програма повідомить про помилку та дасть можливість ввести коректну умову. Після успішного вводу данні що підходять під умову видаляються з таблиці.
4. Change data запрошує данні для конкретної таблиці на які треба змінити існуючі стовпці. Далі запрошується умова за якою будуть вибиратися кортежі що мають змінитися. При введенні некоректних даних (за типом або за умовою) програма повідомить про помилку та дасть можливість ввести коректні данні.
5. Generate data запрошує кількість кортежів, що треба згенерувати. При успішній генерації виводиться повідомлення.
6. Make query запрошує в якій комбінації таблиць буде проведений пошук video + category, users + passport, comment + video + users. Потім запрошується умова за якою буде проведений пошук. Якщо всі данні коректні виводиться результат інакше виводиться помилка і пропонується ввести нові данні.

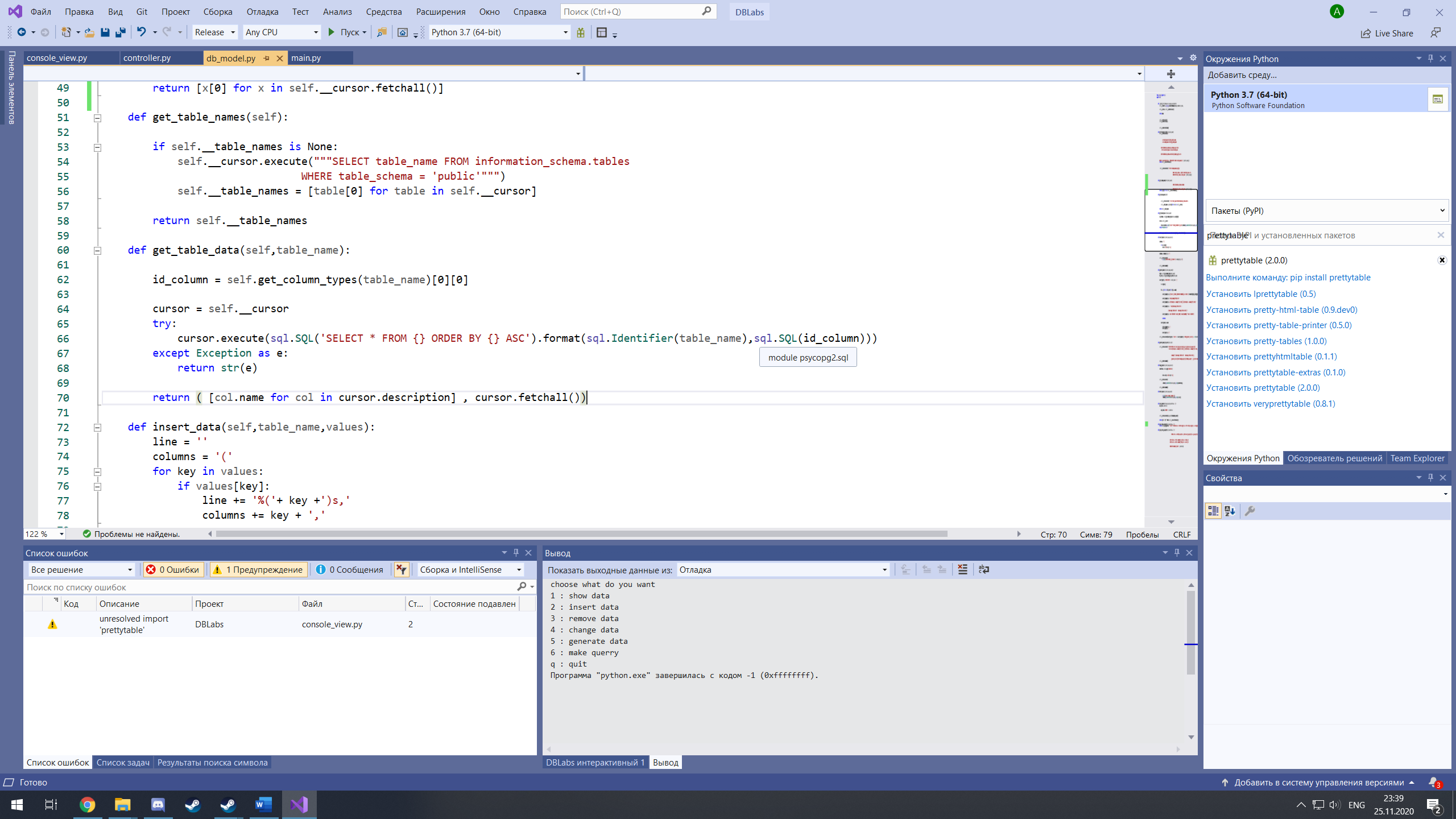
**Вибірка елементів з БД**

Приклад вибірки елементів з таблиці users:



**Рис 2.3 таблиця users**

Метод що повертає вміст таблиці

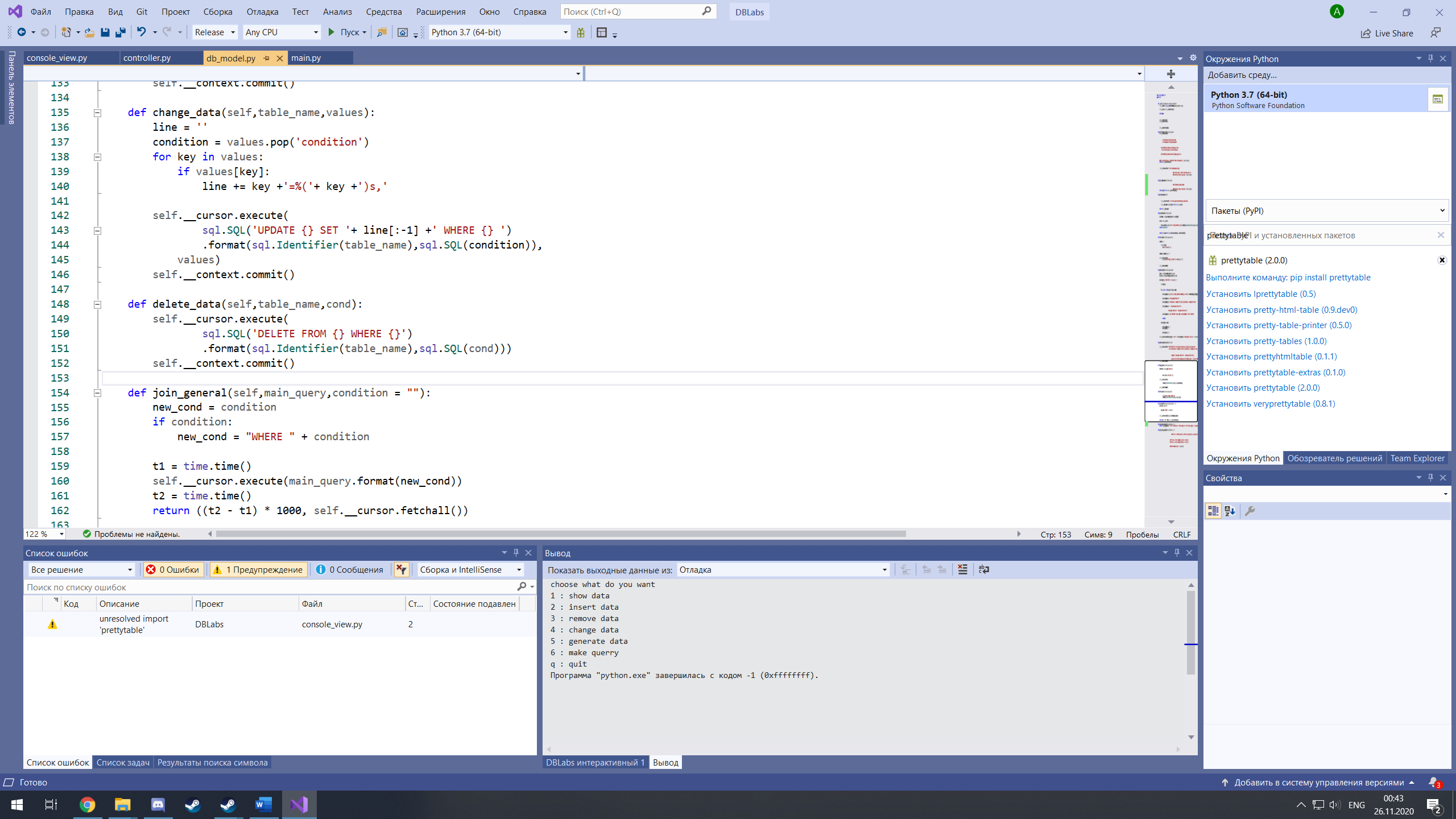


**Рис 2.4** **Метод get\_table\_data**

**Видалення зв’язаних між собою даних**

Наприклад таблиця Users з’єднана з таблицею Passport і Video то при видаленні запису з users будуть видаленні всі пов'язані записи з таблиць passport та video. Це реалізується засобами PostgresSQL за допомогою каскадного видалення.

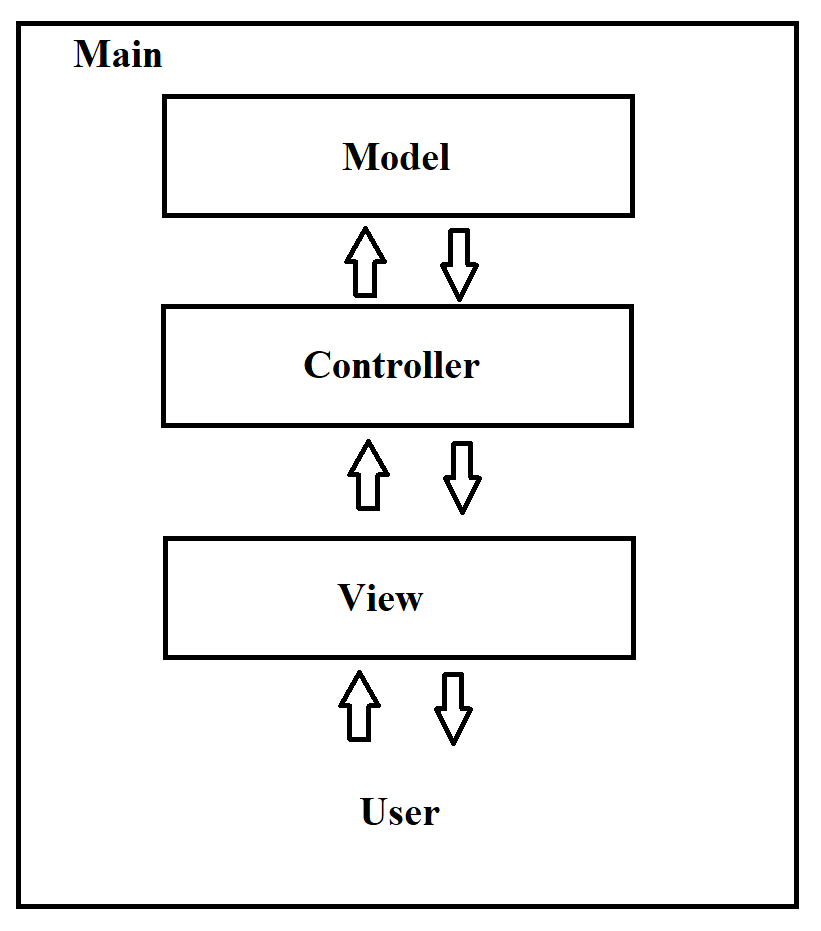
Метод що видаляє данні з таблиці



**Рис 2.4** **Метод delete\_data**

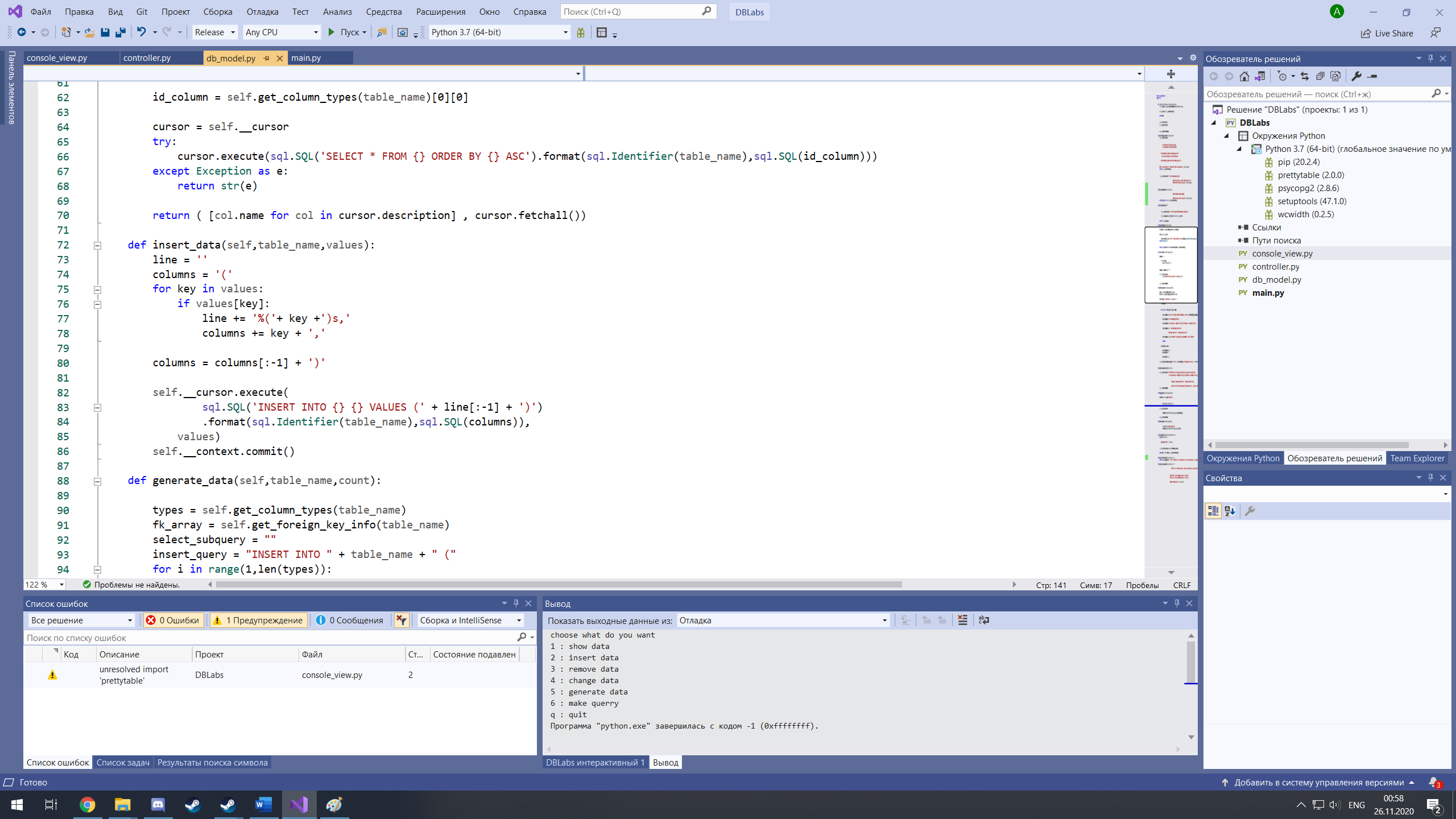
**Структура програми**

Програма створена за патерном MVC (Model-View-Controller). Складається з модулів Controller(controller.py), Model(db\_model.py) View(console\_view.py).

****

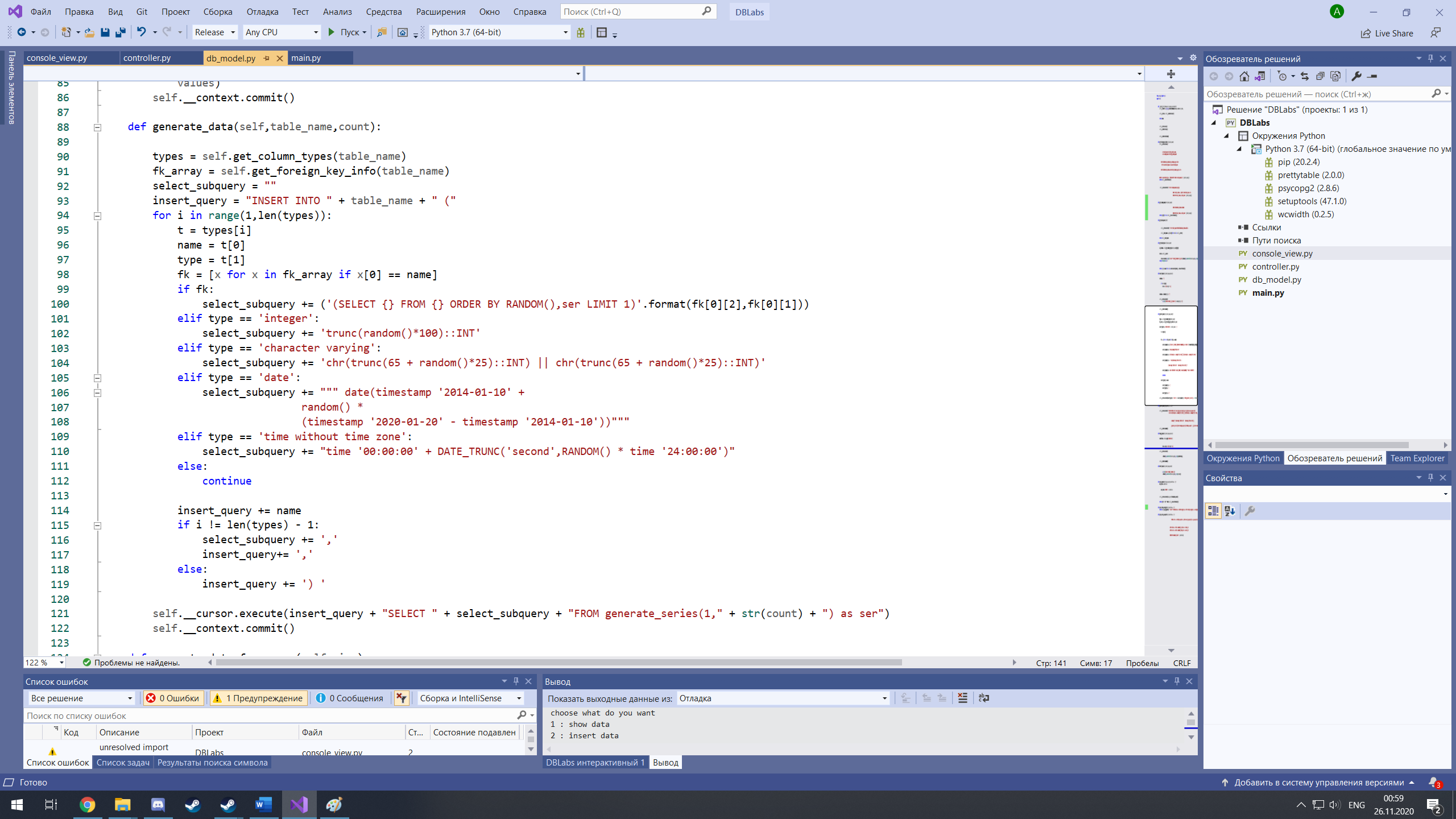
**Рис 2.5 Структура програми**

**Додавання запису в таблицю**



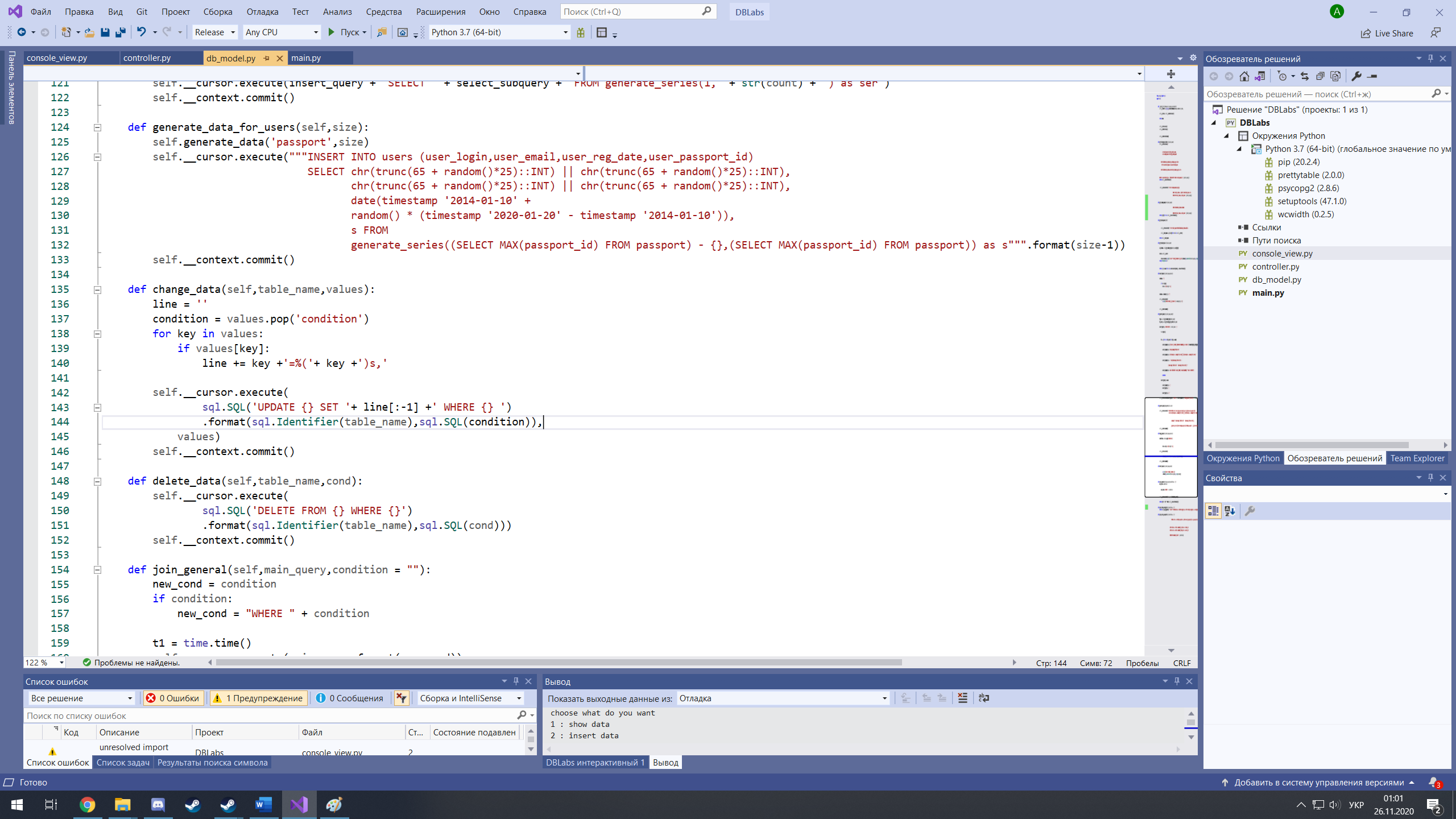
**Рис 2.6** **Метод insert\_data**

**Додавання N сгенерованих записів в таблицю**



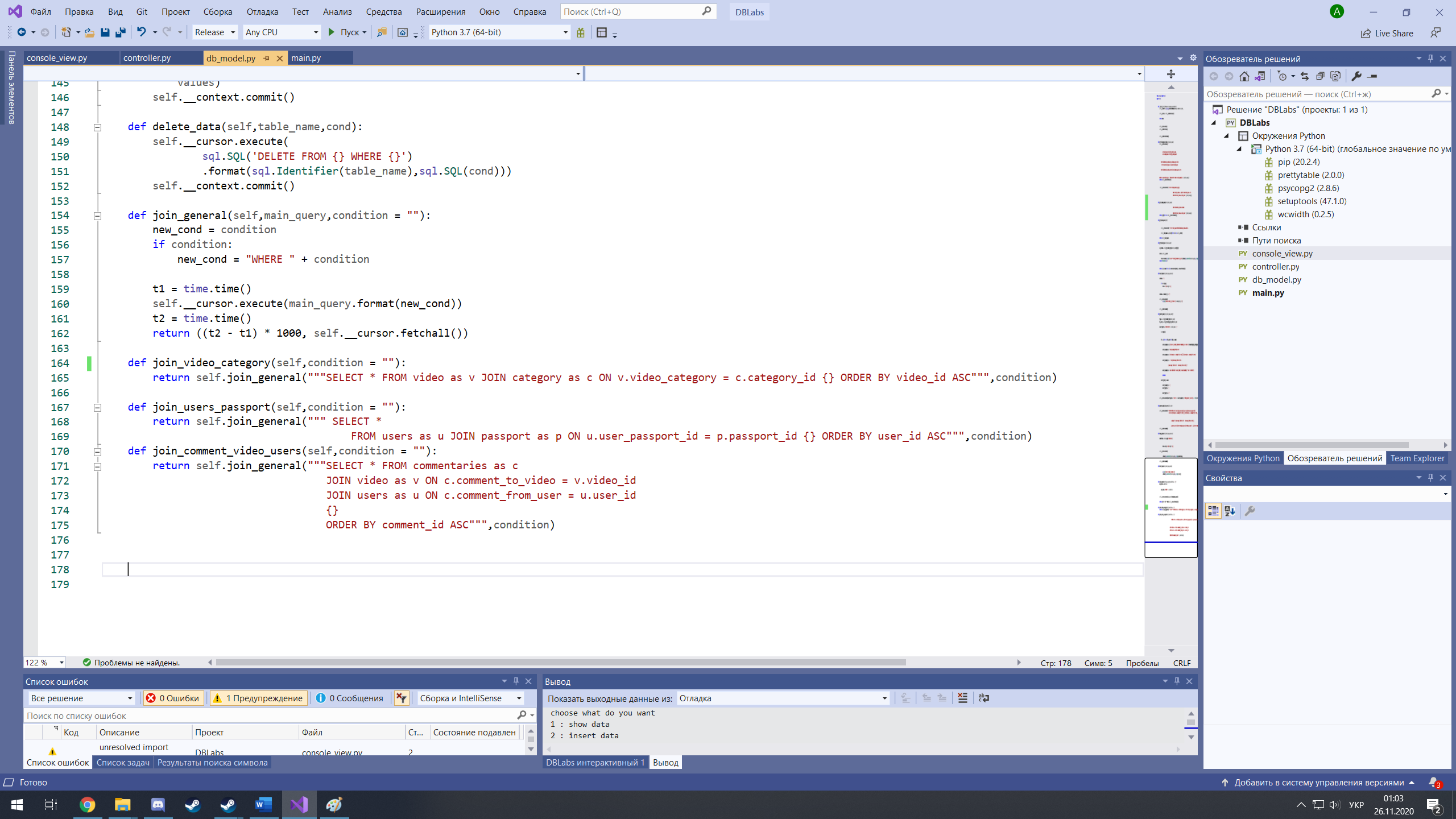
**Рис 2.7** **Метод generate\_data**

**Зміна данних в таблиці**



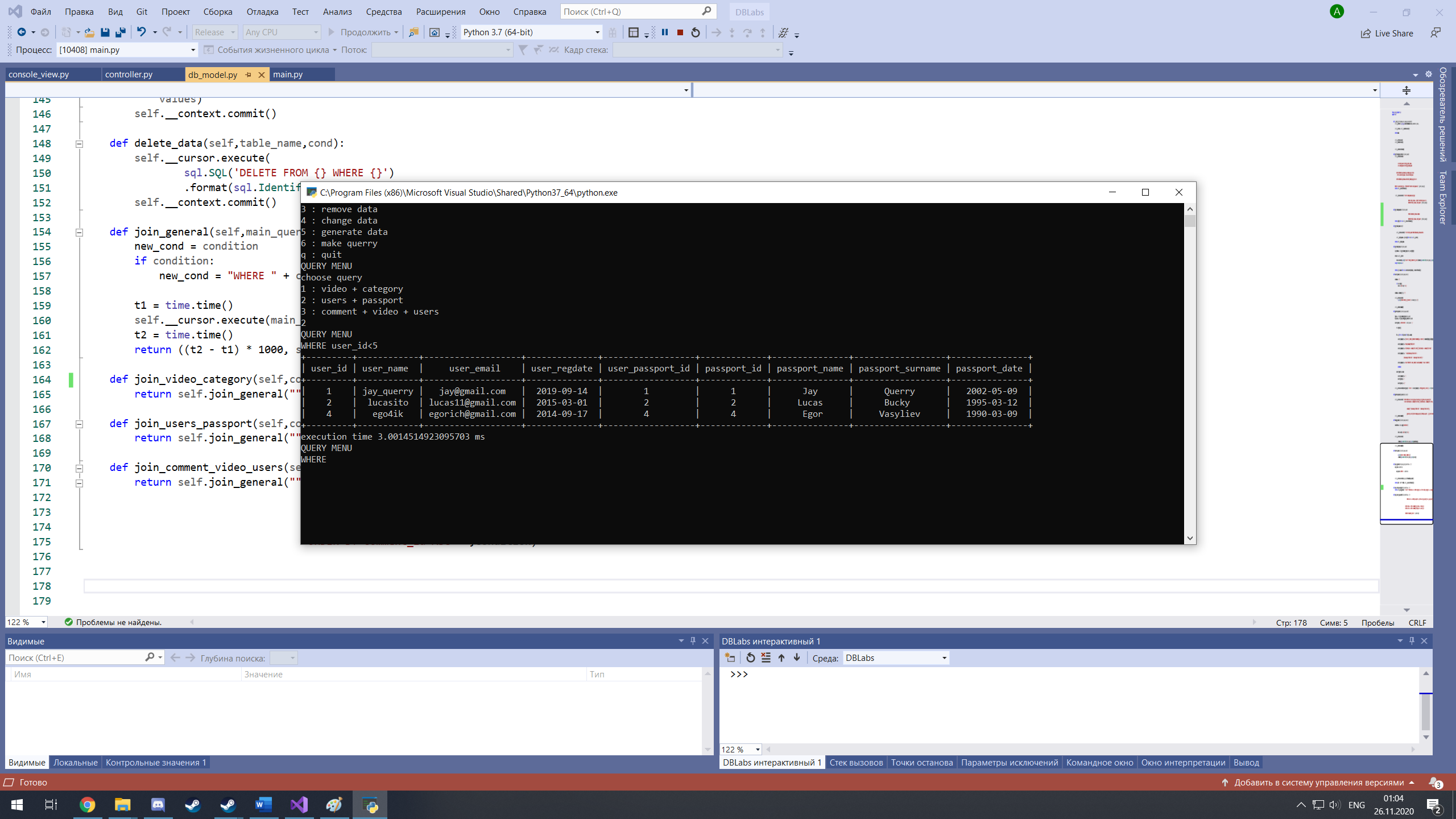
**Рис 2.8** **Метод change\_data**

**Пошук в декількох таблицях**



**Рис 2.9** **Методи join\_general, join\_video\_category, join\_users\_passport, join\_comment\_video\_users**

**Приклад пошуку**



**Рис 2.8** **Об'єднані таблиці users та passport**