Відокремлений структурний підрозділ

«Волинський фаховий коледж

Національного університету харчових технологій»

Звіт до лабораторної роботи №7

“ Розробка ORM-застосунку ”

з дисципліни «**Бази даних**»

Виконав:

Студент 3 курсу, групи ІПЗ-31

**Данилюк Сергій**

Перевірив:

Ройко О.Ю.

Луцьк 2024

**Мета:** Навчитися створювати застосунки, що реалізують технологію ORM.

**Хід роботи**

**Завдання №1 Створення бази даних у DB Browser**

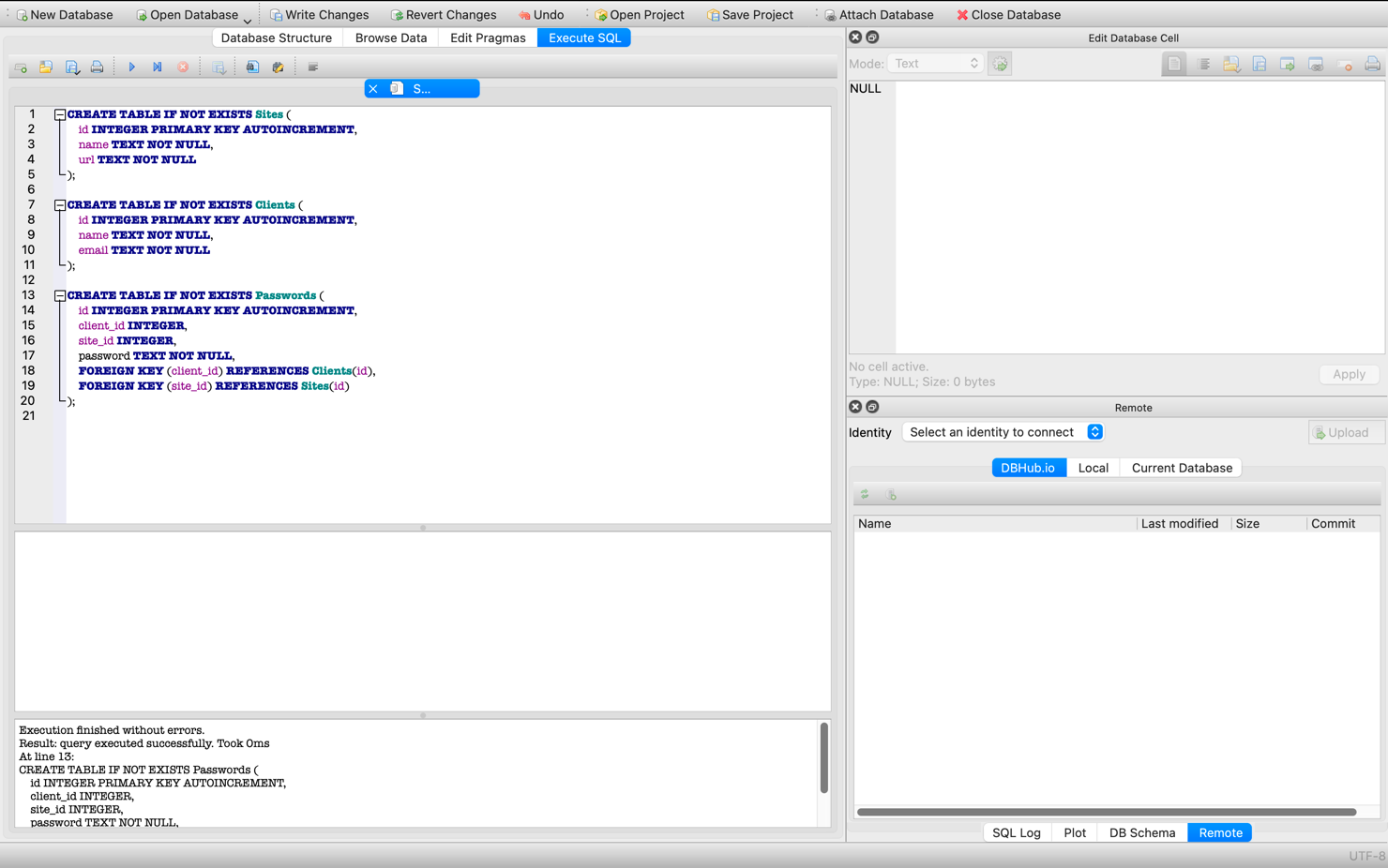
У програмі DB Browser створив нову базу даних із необхідною структурою, згідно з умовами варіанту (Менеджер паролів). Додав таблиці, поля та зв’язки між ними.

1. **Зв'язок між таблицею Passwords і таблицею Clients:**

* Колонка client\_id у таблиці Passwords є зовнішнім ключем, який посилається на колонку id у таблиці Clients.
* Це означає, що кожен запис у таблиці Passwords має бути пов'язаний із записом у таблиці Clients.

1. **Зв'язок між таблицею Passwords і таблицею Sites:**

* Колонка site\_id у таблиці Passwords є зовнішнім ключем, який посилається на колонку id у таблиці Sites.
* Це встановлює зв’язок між записами в таблицях Passwords і Sites.



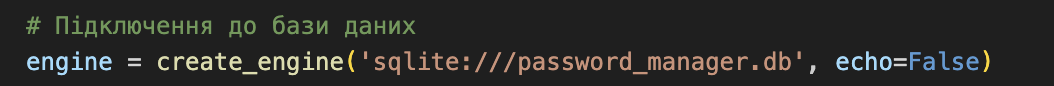
A screenshot of a computer

Description automatically generated

**Завдання №2 Реалізація базових операцій з використанням ORM**

1. **Налаштування бази даних**

1) Підключення до SQLite: База даних створюється автоматично, якщо її ще не існує. У коді це реалізовано через:



SQLite використовується як файлова база даних.

1. **Оголошення моделей:**
2. Sites: таблиця для зберігання сайтів.

A computer screen with text

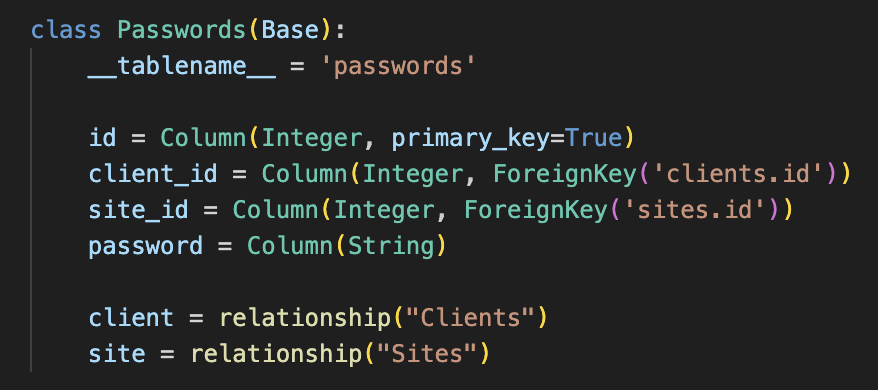
Description automatically generated

1. Clients: таблиця для зберігання клієнтів.

A computer code with text

Description automatically generated

1. Passwords: таблиця для паролів, пов’язана з таблицями Clients і Sites через зовнішні ключі.



1. **Створення таблиць:**

Усі таблиці створюються командою:

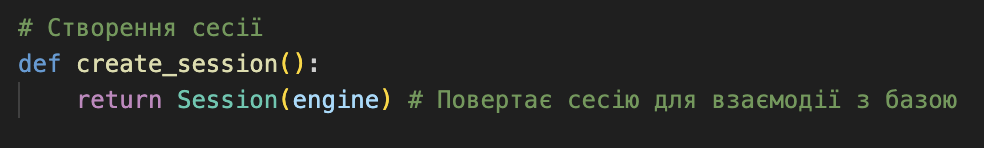
A black and green background with white text

Description automatically generated

**Завдання №3 Створення сесії для роботи з базою**

1. **Створення сесії**

Сесія використовується для взаємодії з базою даних. Для її створення викликається функція:



**Завдання №4 Функції для виконання CRUD-операцій**

1. **Вставка даних:**
2. Додавання сайту: create\_site(session, name, url)

A computer screen with text

Description automatically generated

1. Додавання клієнта: create\_client(session, name, email)

A computer code on a black background

Description automatically generated

1. Додавання пароля: create\_password(session, client\_id, site\_id, password)

**A computer screen with text on it

Description automatically generated**

1. **Читання даних:**
2. Отримання всіх паролів: get\_all\_passwords(session)

A black background with text

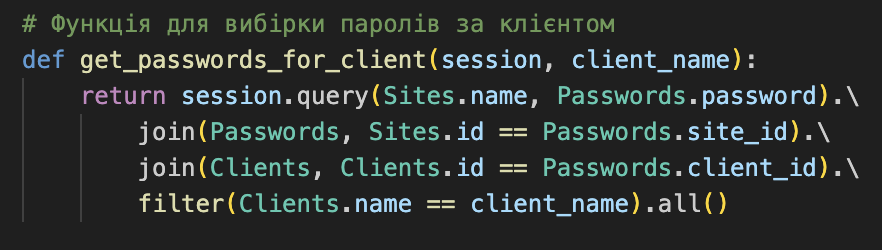
Description automatically generated

1. Отримання паролів з відповідними даними (клієнт, сайт): get\_password\_details(session)

A screen shot of a computer screen

Description automatically generated

1. Отримання паролів за клієнтом: get\_passwords\_for\_client(session, client\_name)



1. **Оновлення даних:**
2. Оновлення пароля: update\_password(session, password\_id, new\_password)

A computer screen with text on it

Description automatically generated

1. **Видалення даних:**
2. Видалення пароля: delete\_password(session, password\_id)

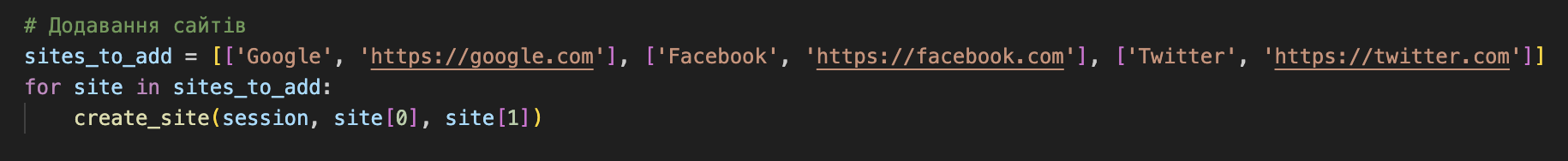
A computer screen with text on it

Description automatically generated

**Завдання №5 Виконання CRUD-операцій**

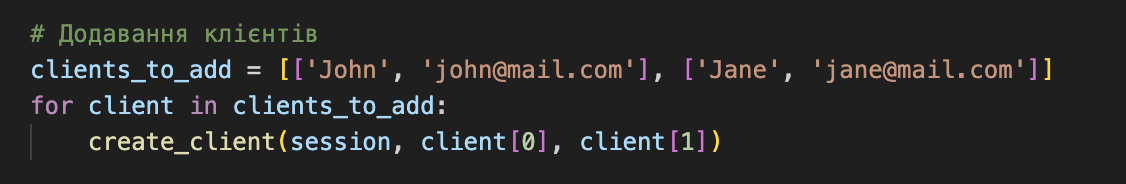
1. Додавання сайтів:

* Кілька сайтів додаються в базу через цикл:



1. Додавання клієнтів:

* Аналогічно додаються клієнти:



1. Додавання паролів:

* Для кожного клієнта додаються паролі до відповідних сайтів:

A computer screen with text

Description automatically generated

1. Читання даних:

* Отримуються всі паролі з деталями (клієнт, сайт, пароль):

A screen shot of a computer code

Description automatically generated

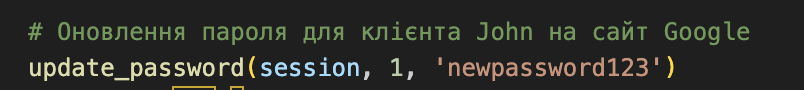
* Отримуються паролі для клієнта "John":

A screen shot of a computer code

Description automatically generated

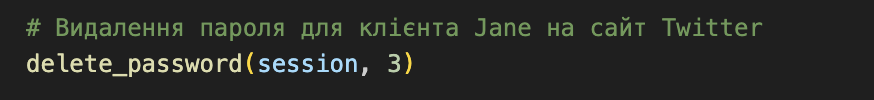
1. **Оновлення пароля:**

* Оновлюється пароль для клієнта "John" на сайті Google:



1. **Видалення пароля:**

* Видаляється пароль для клієнта "Jane" на сайті Twitter:



**Завдання №6 Закриття сесії**

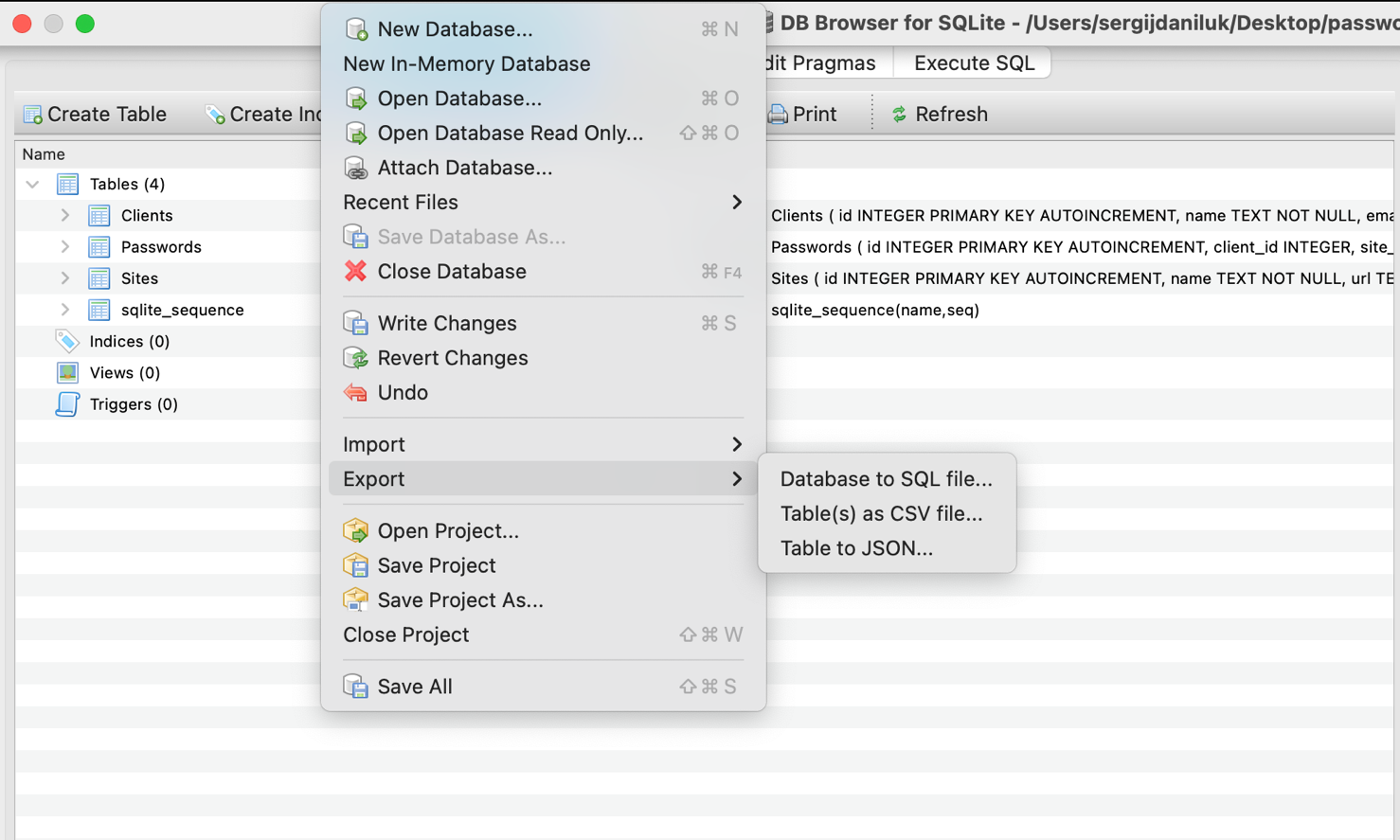
Після завершення роботи сесія закривається через:

A black background with colorful text

Description automatically generated

**Завдання №7 Експорт бази даних**

Після завершення роботи експортував готову базу даних із DB Browser у форматі .sqlite.



A screenshot of a computer

Description automatically generated

**Завдання №8 Створення репозиторію та завантаження проєкту на GitHub**

Посилання на GitHub: <https://github.com/SerhiyD21/PasswordManager_Variant4>

**Висновок:** Під час виконання лабораторної роботи №7 було створено базу даних з використанням інструменту ORM (Object-Relational Mapping) у Python. Були розроблені три взаємопов’язані таблиці: Sites, Clients і Passwords, які забезпечують збереження інформації про сайти, клієнтів та їхні паролі. Було реалізовано основні CRUD-операції: додавання, пошук, оновлення та видалення даних. В процесі виконання завдання я засвоїв принципи створення моделей, налаштування зв’язків між таблицями, а також використання сесій для роботи з базою даних.

**Контрольні запитання (відповіді)**

1. **У чому особливість технології ORM?**

Технологія ORM (Object-Relational Mapping) дозволяє працювати з базами даних на рівні об'єктів у коді, а не на рівні SQL-запитів. Основна особливість ORM у тому, що вона автоматично перетворює записи таблиць у базі даних у об'єкти відповідних класів і навпаки. Це значно спрощує роботу з базами даних, оскільки замість написання складних SQL-запитів можна використовувати методи та властивості об'єктів.

1. **Які переваги має технологія ORM?**

Технологія ORM має багато переваг, які роблять її зручною для розробки програм. Основні переваги:

1. ***Простота використання***

ORM дозволяє працювати з базами даних як із об’єктами в коді, не пишучи складні SQL-запити.

1. ***Швидкість розробки***

Завдяки автоматизації багатьох процесів (наприклад, створення, оновлення чи видалення записів) ORM допомагає швидше реалізувати функціонал програми.

1. ***Менше помилок у коді***

Автоматичне формування SQL-запитів знижує ризик синтаксичних помилок. Також зменшується можливість SQL-ін'єкцій, якщо ORM правильно використовується.

1. ***Можливість роботи з різними СУБД***

ORM-платформи зазвичай підтримують кілька систем управління базами даних (MySQL, SQLite, PostgreSQL тощо). Це дозволяє легко змінювати СУБД без значних змін у коді.

1. ***Підтримка зв’язків між таблицями***

ORM спрощує роботу з реляційними базами даних, автоматично створюючи зв’язки між таблицями через об’єкти.

1. **Яким чином реалізується технологія ORM в Python?**

У Python технологія ORM реалізується за допомогою спеціальних бібліотек, які автоматизують роботу з базами даних. Найпоширенішими є SQLAlchemy та Django ORM. Вони дозволяють працювати з базою даних через об’єкти, замість написання SQL-запитів.

Для реалізації ORM у Python потрібно виконати кілька основних кроків:

***Налаштування підключення до бази даних:*** Спочатку задається підключення до бази даних, використовуючи відповідну бібліотеку. Наприклад, у SQLAlchemy це робиться через строку підключення, яка вказує тип бази даних (SQLite тощо) та шлях до файлу або сервера.

***Створення моделей:*** Таблиці бази даних описуються у вигляді Python-класів, де кожен клас відповідає одній таблиці, а його атрибути — колонкам цієї таблиці. Це дозволяє працювати з даними як з об’єктами.

***Виконання операцій з базою даних:*** ORM автоматично перетворює методи Python-класів у SQL-запити. Наприклад, можна створювати нові записи, змінювати існуючі або видаляти їх за допомогою простих викликів методів.

***Міграції та оновлення структури бази:*** За допомогою ORM можна легко змінювати структуру бази даних, оновлюючи Python-класи. Бібліотека сама генерує відповідні SQL-команди для внесення змін.