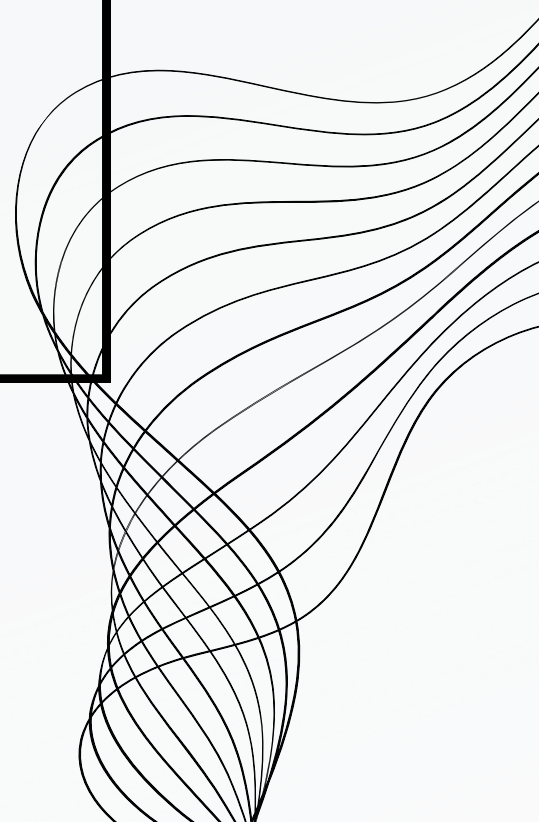




ВВЕДЕННЯ ДО ТЕОРІЇ ХМАРНИХ ОБЧИСЛЕНЬ

**ЛЕКЦІЯ 4. ПРОГРАМУВАННЯ PYTHON ТА SQLITE.
ПРИЗНАЧЕННЯ УТИЛІТИ CSVSQL**



AGENDA

01

ОСНОВНІ ОПЕРАЦІЇ SQL

02

РОБОТА PYTHON З SQLITE

03

ПРИЗНАЧЕННЯ УТИЛІТИ CSVSQL ТА МЕТОД EXECUTE()

4.1. ОСНОВНІ ОПЕРАЦІЇ SQL



SQLite є однією з ефективних реалізацій SQL, яка інтегрується з Python для роботи з базами даних. Використовуючи SQLite, можна створювати з'єднання між Python і базою даних SQL, використовуючи об'єкт підключення та об'єкт курсору для виконання операцій SQL в базі даних.



Мова SQL (Structured Query Language) є стандартною мовою для взаємодії з базами даних і таблицями. Вона має кілька діалектів, але основні операції, такі як створення, читання, оновлення та видалення даних, є стандартними та працюють аналогічно в різних реалізаціях SQL.



SQL складається з трьох мов спеціального призначення: мови визначення даних (DDL) для створення та управління структурою бази даних, мови маніпулювання даними (DML) для додавання, видалення та оновлення даних в таблицях, і мови запитів даних (DQL), яка використовується для отримання даних на основі заданих критеріїв запиту. SQL є важливою мовою для роботи з реляційними базами даних і використовується для розширення можливостей обробки даних в Python.

ДЕЯКІ ЗАГАЛЬНІ КОМАНДИ DATA DEFINITION LANGUAGE SQL

Команда	Пояснення
ALTER TABLE	Змінює структуру існуючої таблиці
CREATE DATABASE	Створює нову порожню базу даних
CREATE TABLE	Створює таблицю в рамках існуючої бази даних
DESCRIBE	Відображає структуру таблиці
DROP DATABASE	Повністю видаляє цілу базу даних
DROP TABLE	Видаляє таблицю з бази даних
USE	Відкриває базу даних, з якою слід працювати

ДЕЯКІ ЗАГАЛЬНІ КОМАНДИ DATA MANIPULATION LANGUAGE SQL

Команда	Пояснення
DELETE	Видаляє наявні дані
INSERT	Додає нові дані
REPLACE	Замінює записи, які мають дублікати даних, на записи, які потрібно вставити
UPDATE	Замінює значення у стовпцях даних новими значеннями залежно від зазначеного критерію

4.2. РОБОТА PYTHON З SQLITE



Розглянемо послідовність команд, які ілюструють основи операцій SQLite. По-перше, в операційну систему потрібно встановити зовнішній інструмент під назвою `csvkit`, щоб файл `csv` можна було імпортувати в базу даних SQLite. Нижченаведені команди відображають етапи створення бази даних SQLite, імпорту даних CSV в базу даних, виконання запиту в таблиці даних та перегляду результатів запиту.

```
import sqlite3 as sql
```

Створюємо нову базу даних та встановлюємо з'єднання з нею:

```
conn = sql.connect('logins.db')
```

```
!csvsql --db sqlite://///logins.db --insert logins.csv
```

Створюємо об'єкт `cursor` для запитів SQL:

```
cur = conn.cursor()
```

4.2. РОБОТА PYTHON З SQLITE



Розглянемо послідовність команд, які ілюструють основи операцій SQLite. По-перше, в операційну систему потрібно встановити зовнішній інструмент під назвою `csvkit`, щоб файл `csv` можна було імпортувати в базу даних SQLite. Нижченаведені команди відображають етапи створення бази даних SQLite, імпорту даних CSV в базу даних, виконання запиту в таблиці даних та перегляду результатів запиту.

Створюємо запит SQL як рядковий об'єкт:

```
query = 'SELECT * FROM logins LIMIT 5'
```

Виконуємо запит SQL, об'єкт `cursor` тепер містить результат запиту:

```
cur.execute(query)
```

Створюємо цикл, який перебирає кількість рядків в об'єкті `cursor` та друкує їх вміст:

```
for row in cur:  
    print(row)
```

4.2. РОБОТА PYTHON З SQLITE

```
(u'John', u'2016-12-29', u'1970-01-01 14:24:13.000000')  
(u'Allan', u'2016-12-29', u'1970-01-01 03:16:54.000000')  
(u'Robert', u'2016-12-30', u'1970-01-01 04:54:25.000000')  
(u'Eve', u'2016-12-30', u'1970-01-01 08:32:14.000000')  
(u'Leslie', u'2016-12-30', u'1970-01-01 20:34:54.000000')
```



Команда "!csvsql --db ..." може бути виконана як перша команда. Це зовнішній інструмент, який потрібно встановити в ОС. Для виконання цього командного рядка можна використовувати командний рядок (Linux CLI), але для спрощення речей цю зовнішню команду можна виконати безпосередньо з ноутбука шляхом префіксації команди "!"

4.3. ПРИЗНАЧЕННЯ УТИЛІТИ CSVSQL



Sqlcsv – простий інструмент командного рядка, який можна використовувати для вибірки дані з бази даних та експорту результат як CSV та вставки даних в базу даних із CSV.

```
CREATE TABLE testtable(  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  int_col INT,  
  float_col FLOAT,  
  varchar_col VARCHAR(255) )
```

4.3. ПРИЗНАЧЕННЯ УТИЛІТИ CSVSQL



Розглянемо, як отримати доступ до бази даних із мови програмування Python. В інших мовах використовується майже однакова модель: імена бібліотек та функцій можуть відрізнятися, але концепції однакові. Ось коротка програма Python, яка вибирає широти та довготи з бази даних SQLite, що зберігається у файлі під назвою `survey.db`:

```
1. import sqlite3
2. connection = sqlite3.connect("survey.db")
3. cursor = connection.cursor()
4. cursor.execute("SELECT Site.lat, Site.long FROM Site;")
5. results = cursor.fetchall()
6. for r in results:
7.     print(r)
8. cursor.close()
9. connection.close()
```

Результат виконання програми:

```
(-49.85, -128.57)
(-47.15, -126.72)
(-48.87, -123.4)
```

ВИСНОВОК ДО ЛЕКЦІЇ 4

SQLite – це реалізація SQL, яка добре працює з Python. Замість того, щоб використовувати метод роботи з клієнтським сервером, використовуються з'єднання, встановлені між Python та базою даних SQL, створюючи об'єкт підключення SQL. Після створення об'єкта з'єднання використовується метод створення об'єкта курсору. Об'єкт курсору має методи SQLite, доступні для виконання операцій SQL в базі даних. SQLite може працювати в інтерактивному режимі з командного рядка, мова Python також може взаємодіяти з SQLite через імпорт модулів.

