

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА

кафедра інформаційних систем та технологій

ЗВІТ

із лабораторної роботи № 7

з дисципліни “**Технології програмування об’єктів лінгвістичної**
предметної галузі”

на тему: “Взаємодія з БД лінгвістичної інформації та текстовими
файлами”

Варіант № **2-05**

Виконав:

Студент групи №2

Кличлієв Кирило

Перевірив:

доц. Костіков М. П.

Київ — 2023

Мета лабораторної роботи

Метою лабораторної роботи №7 є ознайомлення з базами даних, що містять лінгвістичну інформацію, та операціями, які можна проводити над цими даними.

Дані з індивідуального варіанту №2-05

Літера: K

Відмінок: plG

Нове слово: przyjaciel

Рід: 1

Середовище розробки

Мова програмування: Python

Назва та версія IDE: Visual Studio Code 1.78.2

Назва та версія ОС: Linux Mint 21 Vanessa

Мова ОС: англійська

Хід роботи

1. Створюємо новий консольний проєкт (мовою Python або іншою довільною), що при запуску виводить прізвище, ім'я, групу виконавця, а також номер ЛР. Копіюємо в папку з проєктом текстовий файл* parse_lab07.txt (далі polish.txt) і БД* pol_lab07.s3db зі словоформами іменників польської мови.

```
print('Кличлієв Кирило\nГрупа №2\nЛабораторна робота №7')
```

Кличлієв Кирило
Група №2
Лабораторна робота №7

2. Імпортуємо бібліотеку sqlite3, необхідну для виконання поставлених задач.
3. Встановлюємо та перевіряємо з'єднання з БД, виконавши перший SQL-запит: знайти будь-яке одне слово в початковій формі (sgN) та вивести його в консоль.

```
cursor.execute("SELECT sgN FROM tnoun")  
print(f'Перше слово в колонці sgN: {cursor.fetchone()[0]}')
```

Перше слово в колонці sgN: adres

```
cursor.execute("SELECT sgN FROM tnoun WHERE gender IS 2")  
print(f'Перше слово в колонці sgN, що належить до жіночого роду: {cursor.fetchone()[0]}')
```

Перше слово в колонці sgN, що належить до жіночого роду: babcia

4. Створюємо та виконуємо другий SQL-запит: вивести в консоль через кому всі слова, які починаються на задану літеру, в заданому відмінку (за індивідуальним варіантом літера - K, відмінок - plG).

```
cursor.execute("SELECT plG FROM tnoun WHERE plG LIKE 'k_%'")  
k_words = ", ".join([str(word[0]) for word in cursor.fetchall()])  
print(f'Слова з колонки plG, що починаються на літеру K: {k_words}')
```

Результат роботи коду:

Слова з колонки plG, що починаються на літеру K: kamieni, kapeluszy, kapitanów, kawałków, kluczy, kobiet, kochań, kontaktów, końców, koni, korytarzy, kotów, kół, krajów, kroków, królów, krwi, krzesel, krzyków, księży, księżyców, kształtów, kul

5. Створюємо та виконуємо третій SQL-запит, який додає до БД нове слово (за інд. варіантом - przyjaciel) у колонку sgN та його рід (за інд. варіантом - 1) в колонку gender:

```
gender = 1
nw = 'przyjaciel'

cursor.execute(f"INSERT INTO tnoun (gender, sgN) VALUES(\'{gender}\', \'{nw}\')")
cursor.execute("SELECT sgN, gender FROM tnoun ORDER BY id DESC LIMIT 1")
last_word = cursor.fetchone()
print(f'Додане слово: {last_word[0]}\nРід доданого слова: {last_word[1]}')
```

✓ 0.0s

Додане слово: przyjaciel
Рід доданого слова: 1

6. Зчитуємо з файлу “polish.txt” всі граматичні форми однини та множини слова “przyjaciel” в окремі змінні по відмінках: sgN, sgG, sgD, sgA, sgI, sgL, sgV, plN, plG, plD, plA, plI, plL, plV. Мені зручніше працювати з файлами з розширенням .csv і бібліотекою pandas, яка дозволяє легко маніпулювати даними з файлів такого формату. Для конвертації txt-файлу в csv імпортуємо бібліотеку pandas, зчитуємо файл polish.txt і оскільки колонки розділені табуляцією (“\t”), то встановлюємо параметр delimiter=“\t”

```
import pandas as pd

df = pd.read_csv("polish.txt", delimiter='\t')

df.to_csv('polish.csv')
```

Зчитуємо з файлу polish.csv усі граматичні форми заданого слова та зберігаємо їх у словник:

```

words = df[df['punkt'] == 'przyjaciel']

for r in words.itertuples():
    if 'sg:nom' in r[3]:
        sgN = r[1]
    if 'sg:gen' in r[3]:
        sgG = r[1]
    if 'sg:dat' in r[3]:
        sgD = r[1]
    if 'sg:acc' in r[3]:
        sgA = r[1]
    if 'sg:inst' in r[3]:
        sgI = r[1]
    if 'sg:loc' in r[3]:
        sgL = r[1]
    if 'sg:voc' in r[3]:
        sgV = r[1]
    if 'pl:nom' in r[3]:
        plN = r[1]
    if 'pl:gen' in r[3]:
        plG = r[1]
    if 'pl:dat' in r[3]:
        plD = r[1]
    if 'pl:acc' in r[3]:
        plA = r[1]
    if 'pl:inst' in r[3]:
        plI = r[1]
    if 'pl:loc' in r[3]:
        plL = r[1]
    if 'pl:voc' in r[3]:
        plV = r[1]

columns = {'sgN':sgN, 'sgG':sgG, 'sgD':sgD, 'sgA':sgA, 'sgI':sgI, 'sgL':sgL, 'sgV':sgV,
            'plN':plN, 'plG':plG, 'plD':plD, 'plA':plA, 'plI':plI, 'plL':plL, 'plV':plV}

for col, word in columns.items():
    print(f"Відмінок: {col}, словоформа: {word}")

```

✓ 0.1s

```

Відмінок: sgN, словоформа: przyjaciel
Відмінок: sgG, словоформа: przyjaciela
Відмінок: sgD, словоформа: przyjacielowi
Відмінок: sgA, словоформа: przyjaciela
Відмінок: sgI, словоформа: przyjacielem
Відмінок: sgL, словоформа: przyjacielu
Відмінок: sgV, словоформа: przyjacielu
Відмінок: plN, словоформа: przyjaciele
Відмінок: plG, словоформа: przyjaciół
Відмінок: plD, словоформа: przyjaciołom
Відмінок: plA, словоформа: przyjaciół
Відмінок: plI, словоформа: przyjaciółmi
Відмінок: plL, словоформа: przyjaciołach
Відмінок: plV, словоформа: przyjaciele

```

- Створюємо та виконуємо четвертий SQL-запит, який оновлює запис із новим словом у БД, додаючи у відповідні колонки з відмінками всі його граматичні форми.

```
for column, word in columns.items():
    cursor.execute(f"UPDATE tnoun SET {column}='{word}' WHERE sgN='przyjaciel'")

cursor.execute(f"SELECT * FROM tnoun ORDER BY id DESC LIMIT 1")
print(cursor.fetchone())
connection.commit()
```

✓ 0.0s

Результат роботи коду (повертає рядок “przyjaciel” з оновленими даними про інші словоформи аналізованої лексеми):

```
(5644, 1, None, 'przyjaciel', 'przyjaciela', 'przyjacielowi', 'przyjaciela', 'przyjacielem', 'przyjacielu', 'przyjacielu', 'przyjaciele', 'przyjaciół', 'przyjaciółom', 'przyjaciół', 'przyjaciółmi', 'przyjaciółach', 'przyjaciele')
```

Висновки

Отже, під час виконання лабораторної роботи №7 було набуто та закріплено навички роботи з лінгвістичними базами даних. Зокрема, ми, використовуючи бібліотеку sqlite3, підключали БД до пайтонівських скриптів, створювали запити, які дозволяли нам витягувати дані з БД (запит SELECT), а також оновлювати їх (запит UPDATE).