Івано-Франківський національний технічний університет нафти і газу

Кафедра комп’ютерних систем і мереж

**ЛАБОРАТОРНА РОБОТА №10**

з дисципліни Практикум Python

Тема: Робота з базою даних із Python-програми

Виконав:

Ст. гр. КІ-23-2 (Боднар Р.В.)

Перевірив: (Кропивницький Д. Р.)

м. Івано-Франківськ

2025

**Мета роботи:** Ознайомитися з організацією та розробити сховища даних з використанням парадигми ООП.

**Варіант 1**

**Завдання:**

Доповнити розроблену в лабораторній роботі 7 програму збереженням даних в базі даних. Додати до створеного класу методи для запису даних у базу даних, видалення записів із бази, пошуку записів по заданому критерію.

**Хід роботи**

**Код завдання 1:**

from dotenv import load\_dotenv

import psycopg2

import os

from dataclasses import dataclass, asdict

from typing import Dict, List, Optional, Tuple, Any

@dataclass

class Book:

    """Represents a single book."""

    author: str

    title: str

    publisher: str

    genre: str

    year: int

    def \_\_str\_\_(self) -> str:

        return f"{self.title} by {self.author} ({self.year})"

    def to\_dict(self) -> Dict[str, Any]:

        return asdict(self)

class HomeLibrary:

    def \_\_init\_\_(self) -> None:

        self.\_books: Dict[int, Book] = {}

        self.\_next\_id: int = 1

    def init\_db(self, conn):

        with conn.cursor() as cur:

            cur.execute("""

                CREATE TABLE IF NOT EXISTS books (

                    id SERIAL PRIMARY KEY,

                    author TEXT NOT NULL,

                    title TEXT NOT NULL,

                    publisher TEXT,

                    genre TEXT,

                    year INT

                );

            """)

        conn.commit()

    def delete\_from\_db(self, conn, book\_id: int) -> bool:

        with conn.cursor() as cur:

            cur.execute("DELETE FROM books WHERE id = %s RETURNING id;", (book\_id,))

            deleted = cur.fetchone()

        conn.commit()

        return deleted is not None

    def save\_to\_db(self, conn, book: Book) -> int:

        with conn.cursor() as cur:

            cur.execute("""

                INSERT INTO books (author, title, publisher, genre, year)

                VALUES (%s, %s, %s, %s, %s)

                RETURNING id;

            """, (book.author, book.title, book.publisher, book.genre, book.year))

            book\_id = cur.fetchone()[0]

        conn.commit()

        return book\_id

    def search\_in\_db(self, conn, \*\*criteria):

        query = "SELECT id, author, title, publisher, genre, year FROM books WHERE TRUE"

        values = []

        for key, value in criteria.items():

            if key in ["author", "title", "publisher", "genre"]:

                query += f" AND {key} ILIKE %s"

                values.append(f"%{value}%")

            elif key == "year":

                if isinstance(value, tuple) and len(value) == 2:

                    query += " AND year BETWEEN %s AND %s"

                    values.extend(value)

                else:

                    query += " AND year = %s"

                    values.append(value)

        with conn.cursor() as cur:

            cur.execute(query, values)

            rows = cur.fetchall()

        return [(r[0], Book(r[1], r[2], r[3], r[4], r[5])) for r in rows]

    def drop\_table(self, conn):

        with conn.cursor() as cur:

            cur.execute("DROP TABLE IF EXISTS books;")

        conn.commit()

if \_\_name\_\_ == "\_\_main\_\_":

    load\_dotenv()

    conn\_str = os.getenv("DATABASE\_URL")

    conn = psycopg2.connect(conn\_str)

    lib = HomeLibrary()

    lib.init\_db(conn)

    b1 = Book("J.R.R. Tolkien", "The Hobbit", "Allen & Unwin", "Fantasy", 1937)

    b2 = Book("George Orwell", "1984", "Secker & Warburg", "Dystopian", 1949)

    id1 = lib.save\_to\_db(conn, b1)

    id2 = lib.save\_to\_db(conn, b2)

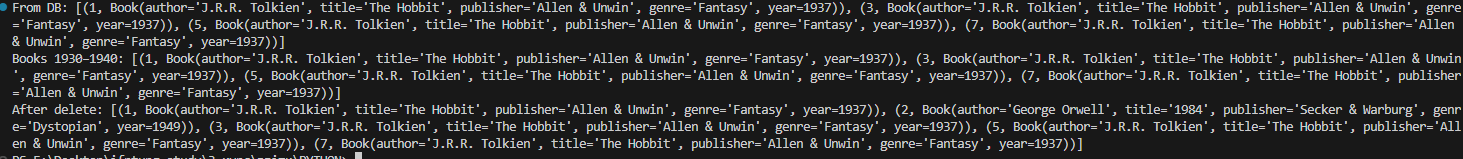
    print("From DB:", lib.search\_in\_db(conn, author="Tolkien"))

    print("Books 1930–1940:", lib.search\_in\_db(conn, year=(1930,1940)))

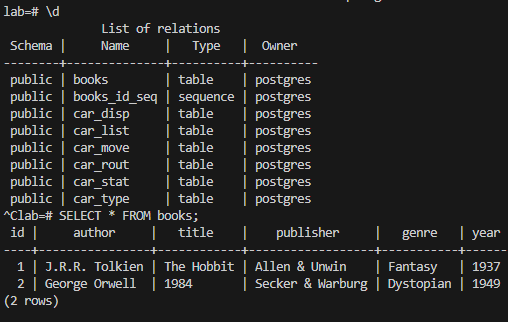
    lib.delete\_from\_db(conn, id2)

    print("After delete:", lib.search\_in\_db(conn))

    #lib.drop\_table(conn)



**Рисунок 1 - Результат роботи програми**

****

**Рисунок 2 - Дані в базі даних**

**Висновок:** У ході виконання лабораторної роботи до створеного раніше класу було додано методи для збереження даних у базі, пошуку за різними критеріями та видалення записів. Реалізація цих методів дозволила забезпечити інтеграцію між об’єктно-орієнтованою структурою програми та реляційною базою даних.