

Міні-проект: Бібліотека

Цей проєкт призначений для практики роботи з базами даних MySQL та мовою Python. Ви створите невелику базу даних для бібліотеки з трьома пов'язаними таблицями, наповните її даними з CSV-файлів та виконаєте кілька запитів.

1. Опис проєкту

Мета: Навчитися створювати та наповнювати реляційну базу даних, а також виконувати складніші запити, що об'єднують дані з кількох таблиць.

Схема: Ми створимо три таблиці:

1. **authors:** Інформація про авторів (ім'я, національність).
2. **genres:** Назви жанрів (поезія, проза тощо).
3. **books:** Інформація про книги (назва), яка буде містити посилання (зв'язки) на authors та genres.

Таблиці books та authors будуть з'єднані за **author_id**, а books та genres — за **genre_id**. Це дозволить нам легко знаходити, наприклад, усі книги певного автора або всі книги певного жанру.

2. Підготовка даних (Створення CSV-файлів)

Перед запуском коду вам потрібно створити три текстові файли у форматі CSV (comma-separated values) та зберегти їх в одній папці зі скриптом.

authors.csv

```
name,nationality
Іван Франко,Україна
Тарас Шевченко,Україна
Леся Українка,Україна
Вільям Шекспір,Велика Британія
Джордж Орвелл,Велика Британія
```

genres.csv

```
name
Поезія
Проза
Драма
Фантастика
Епос
```

books.csv

```
title,author_name,genre_name
Каменярі,Іван Франко,Поезія
Захар Беркут,Іван Франко,Проза
Кобзар,Тарас Шевченко,Поезія
Лісова пісня,Леся Українка,Драма
1984,Джордж Орвелл,Фантастика
Гамлет,Вільям Шекспір,Драма
Ой по горі роман цвіте,Тарас Шевченко,Поезія
Пригоди лиса Микити,Іван Франко,Проза
```

3. Логіка реалізації (Python)

Код, наведений нижче, виконує послідовно наступні кроки:

1. **Підключення до MySQL:** Встановлює з'єднання з вашим MySQL-сервером, використовуючи надані облікові дані.
2. **Створення БД та таблиць:** Запускає SQL-запити (DDL) для створення бази даних `library_db` та трьох таблиць з урахуванням зв'язків. Команда `FOREIGN KEY` створює посилання на інші таблиці.
3. **Завантаження даних:** Читає кожен CSV-файл рядок за рядком. Для кожної книги скрипт спочатку знаходить ідентифікатори (ID) автора та жанру за їхніми назвами, а потім вставляє рядок у таблицю `books`, використовуючи ці ID.
4. **Виконання запитів:** Демонструє, як отримати інформацію, об'єднуючи дані з різних таблиць за допомогою оператора `JOIN`.

4. Повний код проєкту

Нижче наведений повний Python-код, який реалізує всі описані кроки.

```
import mysql.connector
import csv
import os
```

ЧАСТИНА 1: НАЛАШТУВАННЯ ТА СКРИПТИ ДЛЯ ФАЙЛІВ

Перед запуском цього коду, створіть три CSV-файли з такими назвами та вмістом:

- authors.csv

- genres.csv

- books.csv

ЗМІСТ authors.csv:

name,nationality

Іван Франко,Україна

Тарас Шевченко,Україна

Леся Українка,Україна

Вільям Шекспір,Велика Британія

Джордж Орвелл,Велика Британія

ЗМІСТ genres.csv:

name

Поезія

Проза

Драма

Фантастика

Епос

ЗМІСТ books.csv:

title,author_name,genre_name

Каменярі,Іван Франко,Поезія

Захар Беркут,Іван Франко,Проза

Кобзар,Тарас Шевченко,Поезія

Лісова пісня,Леся Українка,Драма

1984,Джордж Орвелл,Фантастика

Гамлет,Вільям Шекспір,Драма

Ой по горі роман цвіте,Тарас Шевченко,Поезія

Пригоди лиса Микити,Іван Франко,Проза

ЧАСТИНА 2: ЛОГІКА ПРОЄКТУ

```
# Конфігурація бази даних
DB_CONFIG = {
    "host": "localhost",
    "user": "root",
    "password": "ВАШ_ПАРОЛЬ" # Введіть ваш пароль тут
}
DB_NAME = "library_db"

def connect_to_db(config, db_name=None):
    """Створює та повертає з'єднання з базою даних."""
    try:
        if db_name:
            config['database'] = db_name
        conn = mysql.connector.connect(**config)
        return conn
    except mysql.connector.Error as err:
        print(f"Помилка підключення до MySQL: {err}")
        return None

def create_database_and_tables(cursor):
```

```

"""Створює базу даних та три таблиці: авторів, жанрів і книг."""
print("Створення бази даних та таблиць...")
try:
    # Створення бази даних
    cursor.execute(f"CREATE DATABASE IF NOT EXISTS {DB_NAME}")
    cursor.execute(f"USE {DB_NAME}")

    # Створення таблиці авторів
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS authors (
            id INT AUTO_INCREMENT PRIMARY KEY,
            name VARCHAR(255) NOT NULL UNIQUE,
            nationality VARCHAR(255)
        );
    """)

    # Створення таблиці жанрів
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS genres (
            id INT AUTO_INCREMENT PRIMARY KEY,
            name VARCHAR(255) NOT NULL UNIQUE
        );
    """)

    # Створення таблиці книг (зв'язок з авторами та жанрами)
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS books (
            id INT AUTO_INCREMENT PRIMARY KEY,
            title VARCHAR(255) NOT NULL,
            author_id INT,
            genre_id INT,
            FOREIGN KEY (author_id) REFERENCES authors(id),
            FOREIGN KEY (genre_id) REFERENCES genres(id)
        );
    """)
    print("База даних та таблиці успішно створені.")

except mysql.connector.Error as err:
    print(f"Помилка при створенні таблиць: {err}")

def load_data_from_csv(conn, cursor):
    """Завантажує дані з CSV-файлів в таблиці."""

```

```

print("Завантаження даних з CSV-файлів...")
try:
    # Вставка авторів
    with open('authors.csv', 'r', encoding='utf-8') as f:
        reader = csv.reader(f)
        next(reader) # Пропускаємо заголовок
        for row in reader:
            name, nationality = row
            try:
                cursor.execute("INSERT INTO authors (name, nationality)
VALUES (%s, %s)", (name, nationality))
            except mysql.connector.IntegrityError:
                print(f"Автор '{name}' вже існує. Пропускаємо...")
        conn.commit()

    # Вставка жанрів
    with open('genres.csv', 'r', encoding='utf-8') as f:
        reader = csv.reader(f)
        next(reader) # Пропускаємо заголовок
        for row in reader:
            name = row[0]
            try:
                cursor.execute("INSERT INTO genres (name) VALUES (%s)",
(name,))
            except mysql.connector.IntegrityError:
                print(f"Жанр '{name}' вже існує. Пропускаємо...")
        conn.commit()

    # Вставка книг
    with open('books.csv', 'r', encoding='utf-8') as f:
        reader = csv.reader(f)
        next(reader) # Пропускаємо заголовок
        for row in reader:
            title, author_name, genre_name = row
            # Отримання ID автора та жанру для зв'язку
            cursor.execute("SELECT id FROM authors WHERE name = %s",
(author_name,))
            author_id = cursor.fetchone()[0]
            cursor.execute("SELECT id FROM genres WHERE name = %s",
(genre_name,))
            genre_id = cursor.fetchone()[0]
            cursor.execute("INSERT INTO books (title, author_id,

```

```

genre_id) VALUES (%s, %s, %s)", (title, author_id, genre_id))
    conn.commit()
    print("Дані успішно завантажено.")

except FileNotFoundError as e:
    print(f"Помилка: Не знайдено файл '{e.filename}'. Переконайтеся,
що файли CSV знаходяться в тій самій папці, що і скрипт.")
except mysql.connector.Error as err:
    print(f"Помилка завантаження даних: {err}")
    conn.rollback()

def run_queries(cursor):
    """Виконує демонстраційні запити."""
    print("\nВиконання запитів:")

    # Запит 1: Знайти всі книги за певним автором
    print("\nКниги Івана Франка:")
    query_ivan_franko = """
        SELECT B.title FROM books B
        JOIN authors A ON B.author_id = A.id
        WHERE A.name = 'Іван Франко';
    """
    cursor.execute(query_ivan_franko)
    for row in cursor.fetchall():
        print(f"- {row[0]}")

    # Запит 2: Знайти всі книги певного жанру
    print("\nКниги в жанрі 'Поезія':")
    query_poetry = """
        SELECT B.title, A.name FROM books B
        JOIN authors A ON B.author_id = A.id
        JOIN genres G ON B.genre_id = G.id
        WHERE G.name = 'Поезія';
    """
    cursor.execute(query_poetry)
    for row in cursor.fetchall():
        print(f"- {row[0]} (Автор: {row[1]})")

    # Запит 3: Знайти всіх авторів з певної країни та їх книги
    print("\nАвтори з України та їхні книги:")
    query_ukrainian_authors = """
        SELECT A.name, B.title FROM authors A
    """

```

```

        JOIN books B ON A.id = B.author_id
        WHERE A.nationality = 'Україна'
        ORDER BY A.name, B.title;
    """
    cursor.execute(query_ukrainian_authors)
    for row in cursor.fetchall():
        print(f"- Автор: {row[0]}, Книга: {row[1]}")

    print("\nЗапити завершено.")

def main():
    """Основна функція для запуску проєкту."""
    conn = None
    try:
        # Підключення до сервера (без вказання БД)
        conn = connect_to_db(DB_CONFIG)
        if not conn:
            return

        cursor = conn.cursor()

        # Створення БД та таблиць
        create_database_and_tables(cursor)
        conn.close() # Закриваємо з'єднання

        # Підключаємось знову, але вже до конкретної БД
        conn = connect_to_db(DB_CONFIG, DB_NAME)
        if not conn:
            return

        cursor = conn.cursor()

        # Завантаження даних
        load_data_from_csv(conn, cursor)

        # Виконання запитів
        run_queries(cursor)

    finally:
        if conn and conn.is_connected():
            print("\nЗакриття з'єднання з MySQL.")
            cursor.close()

```



```
conn.close()
```

```
if __name__ == "__main__":  
    main()
```