

Лабораторна робота № 13

Використання бібліотек Python для опрацювання баз даних.

Мета роботи: оволодіння навичками обробки та зберігання структурованих даних за допомогою словників та csv-файлів.

Короткі теоретичні відомості

В коді **Python** можна використовувати підключення до серверів баз даних і зберігати дані для роботи програми в реляційній (і не тільки) базі даних. Для підключення до бази даних того чи іншого типу слід встановити бібліотеку (драйвер підключення) відповідного типу. Вона міститиме інструменти для підключення до БД та набір функцій і класів для виконання запитів та опрацювання їх результатів. Бібліотеки можна додавати до інтерпретатора Python, встановленого в операційній системі за допомогою спеціального модуля **pip (package installer for Python)**, або додавати до окремого проєкту, створеного в PyCharm.

Опрацювання даних в базі виконується за допомогою SQL-запитів відповідного діалекту. Для їх надсилання серверу потрібно отримати екземпляр об'єкта підключення, вказавши набір параметрів (url сервера, назву БД, ім'я користувача та пароль), після чого можна використовувати його методи для відправки запитів.

Для розміщення бази даних MySQL потрібно встановити MySQL-сервер, але можна скористатися одним з безкоштовних сервісів для тестування MySQL в мережі Інтернет.



Завдання

Розробити інформаційну систему для роботи з даними з предметної області відповідно до варіанта. Програма повинна записувати та читати дані відповідної структури з бази даних, а також надавати можливість редагувати збережені дані та вилучати їх. Реалізувати пошук даних за вказаними у варіанті завдання критеріями.

1. Створити програму для опрацювання БД з даними про авіарейси. Структура даних: номер рейсу, пункт призначення, час відправлення, ціна квитка, кількість вільних місць. В програмі передбачити вивід у вигляді таблиці інформації про всі рейси до заданого користувачем пункту призначення та усі рейси, що відправляються у заданий користувачем проміжок часу.

2. Написати програму для опрацювання БД з інформацією про складання екзаменаційної сесії студентами. Структура даних: шифр групи, прізвище студента, назва предмета, оцінка, назва предмета... (всього 5 предметів). В програмі передбачити вивід списку студентів, які отримали “5” з усіх предметів, а також рейтингу (середньої оцінки з усіх предметів) в упорядкованому за спаданням вигляді.

3. У базі даних записують дані про продаж комп’ютерної техніки наступної структури: код покупця, назва товару, ціна за одиницю товару, продана кількість. Написати програму яка дозволить користувачеві вносити в дані в базу, виводити усі збережені дані на екран (у вигляді таблиці), а також за введенням з клавіатури кодом покупця виводити список куплених ним товарів та загальну вартість покупок, а за введеною назвою товару, – коди покупців, що його придбали та кількість проданих одиниць.

4. Створити програму для роботи з базою даних, що містить дані про рух пасажирських потягів від деякої станції. Запис містить поля: назва кінцевої станції, номер потяга, час відправлення, кількість наявних місць по категоріях вагонів – СВ, купейний, плацкартний, загальний. Програма повинна виводити список потягів за вказаною назвою кінцевої станції та проміжком часу відправлення, а також відбирати лише ті з них, які мають вільні місця вказаної категорії.

5. У базі даних записують дані про продаж турів клієнтам туристичної агенції наступної структури: прізвище та ім’я клієнта, назва туру та його тривалість, вартість туру та відмітка про оплату. Розробити програму яка дозволить користувачеві за введеною з клавіатури назвою туру знайти усіх клієнтів, що придбали цей тур та порахувати їх кількість та загальну суму виручки від продажі турів, а також вивести список усіх клієнтів, що замовили вказаний тур, але не внесли кошти на його оплату.

6. Створити програму для опрацювання бази з даними про оплату комунальних послуг через платіжну систему банку. Структура даних: прізвище платника, адреса платника, вид послуги, дата оплати, сума. В програмі передбачити вивід даних про всі оплати послуг за вказаною адресою а також вивід підсумкової таблиці з сумарною оплатою кожного виду послуг.

7. Створити програму для опрацювання БД з інформацією про запис на прийом протягом поточного тижня відвідувачів поліклініки. Структура даних: прізвище відвідувача, домашня адреса, рік народження, прізвище сімейного лікаря, день тижня, потрібна спеціалізація лікаря. В програмі передбачити вивід даних користувачем та збереження їх у таблиці бази даних, а також вивід у формі таблиці усіх даних з записаного раніше файлу.

8. У базу даних записують інформацію про вакансії, подані ІТ-компаніями, яка містить наступні дані: назва компанії, назва позиції, мова/технологія програмування, вимоги до претендента, орієнтовна заробітна плата. Розробити програму, яка дозволить користувачеві відбирати з бази вакансії за введеною з клавіатури назвою мови програмування та діапазоном для розміру орієнтовної зарплатні, а також переглядати список усіх вакансій, поданих вказаною компанією.

9. Ріелтор зберігає в БД інформацію з оголошень про продаж нерухомості у вигляді записів такої структури: адреса помешкання, кількість кімнат, житлова площа, загальна площа, поверх, ціна, телефон власника. Скласти програму яка дозволить користувачеві шукати та виводити на екран інформацію за такими параметрами: за введеною з клавіатури кількістю кімнат та діапазоном цін, або за кількістю кімнат та орієнтовною загальною чи житловою площею помешкання.

10. Створити програму для роботи з базою даних, що містить дані про продаж вживаних автомобілів. Запис містить поля: марка авто, модель авто, тип кузова, об'єм двигуна, рік випуску, ціна. Програма повинна виводити список авто за такими вказаними користувачем параметрами: марка, модель та діапазон цін або марка, модель та період випуску (у роках, від... до...).

11. Рекрутер записує в БД інформацію щодо резюме, розміщених ІТ-фахівцями у сервісах пошуку роботи. Записи містять інформацію такої структури: прізвище та ім'я, електронна адреса, місто проживання, перелік технологій (текст, в якому перераховані технології відділяються комами), досвід роботи, бажання працювати віддалено, можливість зміни міста проживання. Розробити програму яка дозволить йому відбирати для офісу із заданого міста резюме, що відповідають вказаній технології, розділивши їх на три частини: претенденти, що проживають у заданому місті, претенденти, готові до віддаленої роботи та претенденти, що погодяться переїхати у вказане місто.

12. Менеджер паркінгу реєструє в базі даних інформацію про парковку авто та оплату послуг їх власниками. Записи містять поля: марка та модель авто, державний номерний знак, телефон власника, година парковки (ціле

число) та кількість годин, за яку вноситься передоплата. Скласти програму для пошуку даних, яка дозволить менеджеру, вказавши поточний час (годину) дізнатися список авто, власникам яких доведеться доплатити за парковку та за скільки годин, а також дізнатися, чи авто може без доплати покинути паркінг, вказавши його номер.

13. Створити програму для опрацювання БД з даними про міжміські автобусні рейси. Запис містить поля: назва міста призначення, час відправлення, вартість квитка, загальна кількість місць в автобусі, кількість проданих квитків. Запрограмувати визначення заповненості (у %) рейсів до вказаного міста чи обчислення сумарної виручки для кожного рейсу (за вибором користувача програми) та вивід результатів у вигляді таблиці.

14. Менеджер онлайн-магазину зберігає дані про замовлення, що надходять. Структура даних: повна назва товару, ціна, кількість, прізвище та ім'я замовника, населений пункт для доставки, спосіб доставки, здійснення оплати. Запрограмувати вивід з бази даних усіх оплачених замовлень, які слід доставити у зазначене місто, та вивід усіх неоплачених замовлень за заданим способом доставки товару.

15. Дані про доступні для перегляду на онлайн-сервісі фільми зберігаються у базі даних у вигляді записів такої структури: назва фільму, жанр, рік виходу на екрани, назва студії, прізвище режисера, загальна тривалість та кількість серій. Скласти програму для пошуку фільмів, яка за вказаним жанром виводитиме на екран два списки окремо одно-, а окремо багатосерійні фільми цього жанру, а також за вказаною назвою кінокомпанії та діапазоном років виводити інформацію про фільми, зняті компанією у цей період.

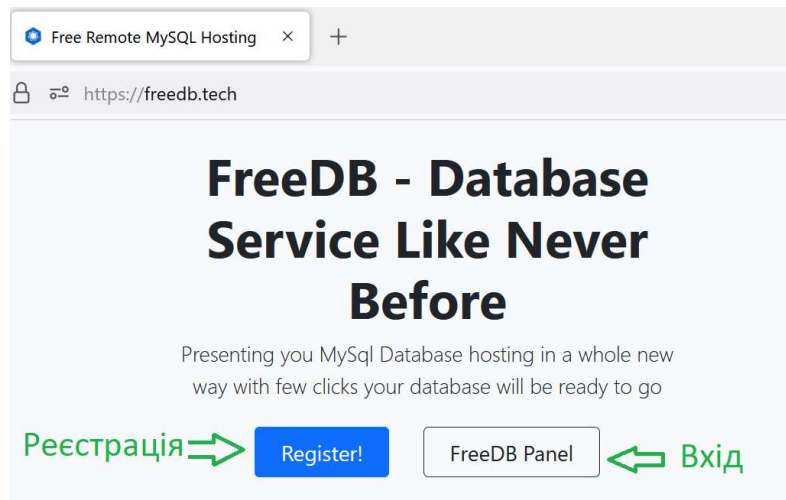
16. Створити програму для опрацювання бази даних про наявність книг у книжковому магазині. Запис містить поля: назва видавництва, назва книги, прізвище автора, рік видання, роздрібна ціна, кількість примірників на складі. Запрограмувати пошук книг за прізвищем автора чи назвою (або її частиною), а також вивід списку усіх книг, які вийшли у вказаному видавництві.

Приклад виконання завдання 1.

Варіант № 16. Умова задачі: Розробити інформаційну систему для роботи з даними з предметної області відповідно до варіанта. Програма повинна записувати та читати дані відповідної структури з бази даних, а також надавати можливість редагувати збережені дані та вилучати їх. Реалізувати пошук даних за вказаними у варіанті завдання критеріями.

16. Створити програму для опрацювання бази даних про наявність книг у книжковому магазині. Запис містить поля: назва видавництва, назва книги, прізвище автора, рік видання, роздрібна ціна, кількість примірників на складі. Запрограмувати пошук книг за прізвищем автора чи назвою (або її частиною), а також вивід списку усіх книг, які вийшли у вказаному видавництві.

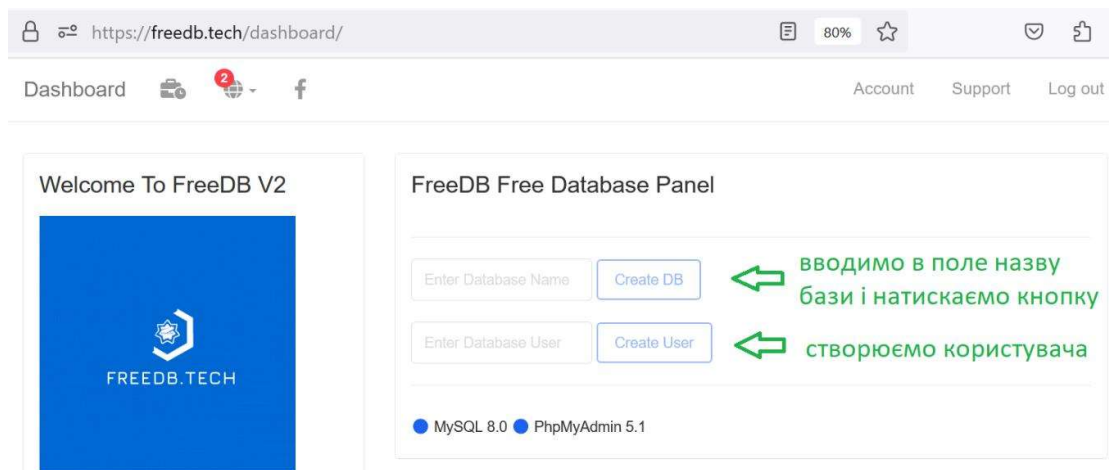
Використовуватимемо базу даних MySQL. Для розміщення бази даних на сервері скористаємося одним з безкоштовних сервісів з хостингу БД MySQL, наприклад, <https://db4free.net/> або <https://freedb.tech/>. Спочатку потрібно зареєструватися в сервісі та створити базу даних. Для реєстрації вказуємо власний e-mail та задаємо пароль для доступу до панелі керування базою даних.



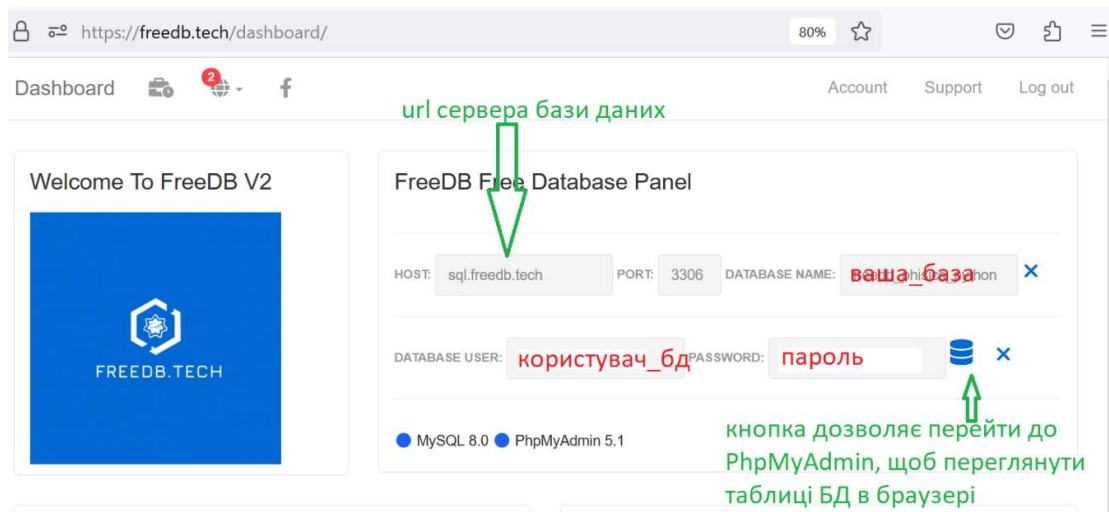
Для реєстрації на <https://freedb.tech/> потрібно також вказати мету, з якою створюється база даних (наприклад, Education and training). Перед створенням бази даних слід підтвердити email-адресу за допомогою отриманого від сервісу електронного листа (зверніть увагу, що деколи такі листи можуть потрапляти до папки зі спамом).

Далі потрібно створити базу даних і користувача для підключення до неї (пароль система генерує сама).

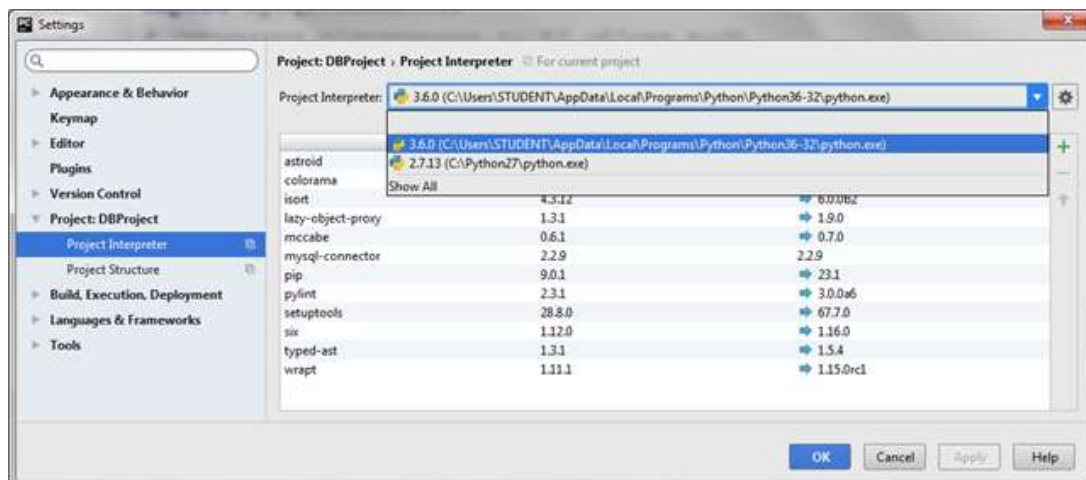
Лабораторний практикум з програмування на Python



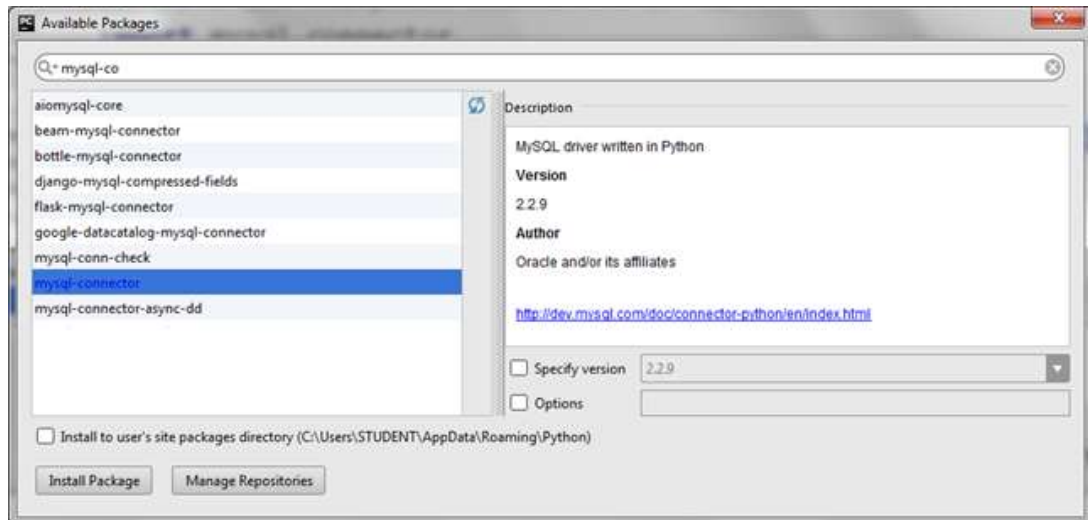
Ці дані будуть використовуватися в коді програми. Їх можна скопіювати з панелі керування, зайшовши у створений обліковий запис.



Після створення бази даних переходимо до розробки проєкту на Python. Запускаємо **Pycharm** та створюємо новий проєкт. Насамперед підключаємо до проєкту модуль з драйвером бази даних **mysql-connector**. Для цього потрібно виконати команду **Settings...** з меню **File**.



У вікні Settings відкриваємо вузол створеного проєкту та вибираємо пункт Project Interpreter, щоб побачити список підключених бібліотек. Далі натискаємо кнопку з позначкою «+» та виконуємо пошук модуля **mysql-connector**, набравши у відповідному полі його назву.



Рекомендовану версію знайденого пакету можна залишити. Натискаємо кнопку **Install Package**. Після успішного встановлення пакету переходимо до написання програмного коду, створивши в проєкті новий Python-файл.

Спочатку перевіримо можливість підключення до бази даних за допомогою такого коду (параметри **користувач_бд** та **пароль** слід заповнити своїми даними):

```
# Модуль для створення підключень до БД
import mysql.connector

# Отримання підключення до БД об'єкт mydb
mydb = mysql.connector.connect(
    host="sql.freedb.tech",
    user="користувач_бд",
    password="пароль"
)
# Друк об'єкта бази даних mydb
print(mydb)
```

Мета поданого вище коду отримати об'єкт підключення до бази даних, результат його коректної роботи виглядатиме приблизно так:

```
E:\PythonScripts\Lab14Projects\venv\Scripts\python.exe
E:\PythonScripts\Lab14Projects\BooksIS.py
```

```
<mysql.connector.connection.MySQLConnection object at 0x0146F448>
```

```
Process finished with exit code 0 підключення до бази даних
```

У випадку некоректної роботи цього фрагмента слід ще раз перевірити правильність виконання усіх попередніх кроків (створення БД, інсталяція драйвера, параметри підключення до БД). Продовжувати написання коду в такому випадку нема сенсу.

Наступним кроком реалізуємо простий сценарій створення тестової таблиці БД, внесення в неї деяких даних, з почальшим їх читанням з бази:

```
# Модуль для створення підключень до БД
import mysql.connector
# Отримання підключення до БД об'єкт mydb
mydb = mysql.connector.connect (
    host="sql.freedb.tech",
    user="користувач_бд",
    password="пароль",
    database = "ваша_база"
)
# Отримання курсора для виконання запитів
mycursor = mydb.cursor()

mycursor.execute('CREATE TABLE IF NOT EXISTS '
                 'MyTestTable(TestId Int, TestName VARCHAR(50))')
mycursor.execute("INSERT INTO MyTestTable "
                 "VALUES(1, 'Test1'), (33, 'Supertest')")
mydb.commit()

mycursor.execute("SELECT * FROM MyTestTable")

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

У результаті роботи цього коду в БД буде створено таблицю MyTestTable. Лише перший раз, якщо таблиця вже створена, дані будуть записуватися в неї. В цю таблицю будуть додані два записи (1, Test) та (33, Supertest), причому стільки разів, скільки буде виконано цей скрипт. На завершення усі внесені записи читаються з бази та виводять ся на екран.

Далі розширимо написаний фрагмент для створення таблиці БД та функції для читання-запису в неї списку книг. Код програми:

```
# Модуль для створення підключень до БД
import mysql.connector
# Функція для створення підключення до БД об'єкт mydb
def get_connection():
    connection = mysql.connector.connect(
        host="sql.freedb.tech",
        user="користувач_бд",
        password="пароль",
        database = "ваша_база"
    )
    return connection

# функція для вводу даних про окрему книжку
# формує і повертає словник з даними
def input_book():
    book_title = input('Назва книги: ')
    authors = input('Автори: ')
    publisher = input('Видавництво: ')
    publish_year = int(input('Рік видання: '))
    price = float(input('Ціна: '))
    count = int(input('Кількість: '))
    book = {'book_title' : book_title, 'authors':
        authors, 'publisher' : publisher,
        'publish_year': publish_year, 'price' : price,
        'count' : count}
    return book

# функція запису даних про нову книгу
# за відсутності таблиці створюватиме нову таблицю
def write_book(book):
    # Отримання курсора для виконання запитів
    mydb = get_connection()
    mycursor = mydb.cursor()
    # Запит створює нову таблицю лише у випадку,
    # якщо її нема в базі даних
    mycursor.execute('CREATE TABLE IF NOT EXISTS Books '
        '(Id Int NOT NULL AUTO_INCREMENT, '
        'Title VARCHAR(250), Authors '
        'VARCHAR(100), Publisher VARCHAR(50), '
        'PublishYear INT, Price DOUBLE, '
        'Count INT, PRIMARY KEY (Id));')
    query = ("INSERT INTO Books (Title, Authors, "
        "Publisher, PublishYear, Price, Count) "
```

Лабораторний практикум з програмування на Python

```
        "VALUES(%s, %s, %s, %s, %s, %s)")
values = (book['book_title'], book['authors'],
          book['publisher'], str(book['publish_year']),
          str(book['price']), str(book['count']))
mycursor.execute(query, values)
mydb.commit()

# зчитування даних про книги та вивід їх на консоль
def read_all():
    # Отримання курсора для виконання запитів
    mydb = get_connection()
    mycursor = mydb.cursor()
    mycursor.execute("SELECT * FROM Books")
    for row in mycursor:
        print(row)
    mycursor.close()

# функція для видалення запису з бази за ключем Id
#
def delete_book():
    # Отримання курсора для виконання запитів
    mydb = get_connection()
    mycursor = mydb.cursor()
    id_for_del = int(input('Id книги для видалення: '))
    query = "DELETE FROM Books WHERE Id = %s"
    id_str = (str(id_for_del),)
    mycursor.execute(query, id_str)
    mydb.commit()

# Основна програма перевіряє роботу створених функцій
new_book = input_book()
write_book(new_book)
read_all()
delete_book()
read_all()
```

Перенесемо створені функції в окремий Python-файл (модуль). Назвемо його, наприклад **booksmysql.py**. Створимо в модулі функції для пошуку даних відповідно до умови задачі: за назвою, прізвищем автора або видавництвом. Код модуля:

```
# Модуль для роботи зі мписком книг у базі даних MySQL
# імпорт модуля для створення підключень до БД
import mysql.connector
# Функція для створення підключення до БД об'єкт mydb
def get_connection():
    connection = mysql.connector.connect(
        host="sql.freedb.tech",
        user="користувач_бд",
        password="пароль",
        database = "ваша_база"
    )
    return connection

# функція для вводу даних про окрему книжку
# формує і повертає словник з даними
def input_book():
    book_title = input('Назва книги: ')
    authors = input('Автори: ')
    publisher = input('Видавництво: ')
    publish_year = int(input('Рік видання: '))
    price = float(input('Ціна: '))
    count = int(input('Кількість: '))
    book = {'book_title' : book_title, 'authors':
        authors, 'publisher' : publisher,
        'publish_year': publish_year, 'price' : price,
        'count' : count}
    return book

# функція запису даних про нову книгу
# за відсутності таблиці створюватиме нову таблицю
def write_book(book):
    # Отримання курсора для виконання запитів
    mydb = get_connection()
    mycursor = mydb.cursor()
    # Запит створює нову таблицю лише у випадку,
    # якщо її нема в базі даних
    mycursor.execute('CREATE TABLE IF NOT EXISTS Books '
        '(Id Int NOT NULL AUTO_INCREMENT, '
        'Title VARCHAR(250), Authors '
        'VARCHAR(100), Publisher VARCHAR(50), '
        'PublishYear INT, Price DOUBLE,')
```

Лабораторний практикум з програмування на Python

```
        'Count INT, PRIMARY KEY (Id));')
query = ("INSERT INTO Books (Title, Authors, "
        "Publisher, PublishYear, Price, Count) "
        "VALUES(%s, %s, %s, %s, %s, %s)")
values = (book['book_title'], book['authors'],
        book['publisher'], str(book['publish_year']),
        str(book['price']), str(book['count']))
mycursor.execute(query, values)
mydb.commit()

# зчитування даних про книги та вивід їх на консоль
def read_all():
    # Отримання курсора для виконання запитів
    mydb = get_connection()
    mycursor = mydb.cursor()
    mycursor.execute("SELECT * FROM Books")
    for row in mycursor:
        print(row)
    mycursor.close()

# функція для видалення запису з бази за ключем Id
#
def delete_book(id_for_del):
    # Отримання курсора для виконання запитів
    mydb = get_connection()
    mycursor = mydb.cursor()
    query = "DELETE FROM Books WHERE Id = %s"
    id_str = (str(id_for_del),)
    mycursor.execute(query, id_str)
    mydb.commit()

# пошук даних про книги за прізвищем автора
# та вивід їх на консоль
def read_by_author(author):
    # Отримання курсора для виконання запитів
    mydb = get_connection()
    mycursor = mydb.cursor()
    query = "SELECT * FROM Books " \
            "WHERE Authors Like %s"
    # враховуємо, що авторів може бути декількв
    args = ('%'+author+'%',)
    mycursor.execute(query, args)
    for row in mycursor:
        print(row)
    mycursor.close()
```

```

# пошук даних про книги за назвою книги,
# або її частиною та вивід їх на консоль
def read_by_title(name):
    # Отримання курсора для виконання запитів
    mydb = get_connection()
    mycursor = mydb.cursor()
    query = "SELECT * FROM Books " \
            "WHERE Title Like %s"
    # враховуємо, що авторів може бути декількв
    args = ('%'+name+'%',)
    mycursor.execute(query, args)
    for row in mycursor:
        print(row)
    mycursor.close()

```

```

# пошук даних про книги за назвою видавництва,
# та вивід їх на консоль
def read_by_publisher(publisher):
    # Отримання курсора для виконання запитів
    mydb = get_connection()
    mycursor = mydb.cursor()
    query = "SELECT * FROM Books " \
            "WHERE Publisher Like %s"
    # враховуємо, що авторів може бути декількв
    args = ('%'+publisher+'%',)
    mycursor.execute(query, args)
    for row in mycursor:
        print(row)
    mycursor.close()

```

В основній програмі підключаємо створений модуль та організовуємо вічний цикл з вибором користувачем відповідної опції роботи програми:

```

# Підключення модуля з функціями для роботи з БД
import booksmysql

# Основна програма
choise = 7 # ненульове значення для входу в цикл
# за умовою в меню 0 - вихід з циклу, і цикл не
# виконається жодного разу, якщо ініціалізувати цю
# змінну значенням 0, а оголошення та
# ініціалізація змінної choise потрібні до циклу
while choise != 0:
    # вивід меню програми

```

Лабораторний практикум з програмування на Python

```
print('Оберіть потрібну дію:')
print('1 - дописати нову книгу в базу')
print('2 - переглянути всі книги в базі')
print('3 - пошук книг за автором')
print('4 - пошук книг за назвою')
print('5 - пошук книг за видавництвом')
print('6 - вилучити книгу з бази')
print('0 - завершити роботу програми')
# зчитування пункту меню обраного користувачем
# та виконання відповідної дій з даними
choise = int(input('>>>'))
if choise == 1:
    booksmysql.write_book(booksmysql.input_book())
elif choise == 2:
    booksmysql.read_all()
elif choise == 3:
    test = input('Введіть прізвище автора >> ')
    booksmysql.read_by_author(test)
elif choise == 4:
    test = input('Введіть назву книги >> ')
    booksmysql.read_by_title(test)
elif choise == 5:
    test = input('Введіть назву видавництва >> ')
    booksmysql.read_by_publisher(test)
elif choise == 6:
    id = int(input('Введіть id книги >> '))
    booksmysql.delete_book(id)
```