

## Кросплатформне програмування Python

### Змістовий модуль 2.

Прикладне програмування мовою Python: від обробки даних до веб-розробок

### Лабораторна робота № 9

#### Корисні посилання

1. Docker: Accelerated Container Application Development <https://www.docker.com/>
2. Docker Hub Container Image Library <https://hub.docker.com/>
3. Найкращі бібліотеки Python для створення баз даних SQL  
<https://eternalhost.net/blog/razrabotka/python-mysql-postgresql-sqlite>
4. Welcome to Python.org <https://www.python.org/>
5. PyCharm: IDE для професійної розробки на Python <https://www.jetbrains.com/ru-ru/pycharm/>

Робота повинна складатися з текстового звіту, що завантажений на eLearn і проекту, що завантажений на GitHub.

#### Структура звіту:

- титульний аркуш (приклад наведено нижче),
- текст завдання,
- скріншоти з назвами використаних в проекті docker образів та docker контейнерів,
- текст Python коду програм і SQL запитів,
- скріншоти виконання програм в власному віртуальному середовищі
- скріншоти всіх таблиць і запитів БД в графічному клієнті
- посилання на проект на GitHub

#### Завдання:

- На платформі Docker, за допомогою файлу docker-compose.yml, створити контейнер з СУБД PostgreSQL або MySQL. Зробити прокидання портів та папок для зберігання БД.
- В цьому контейнері, використовуючи мову Python, створити базу даних. Створити в ній необхідні таблиці з відповідними полями і заповнити їх даними (*вимоги, опис БД та мінімальний об'єм даних, який необхідно ввести в таблиці, вказані в кожному варіанті*).
- На мові Python написати програму, що підключається до створеної БД, виводить всі таблиці (структурата + дані, які в ній зберігаються) та результати виконання запитів в консоль в форматованому вигляді (заголовки стовпців + всі стовпці рівні).
- На платформі Docker створити контейнер з графічним клієнтом (адмінка) для управління БД. Запустити його і підключитись до створеної БД. Переконатись, що всі таблиці і запити створені вірно.
- Завантажити проект на GitHub, попередньо додавши до файлу .gitignore всі технічні папки та файли.
- В текстовому редакторі створити звіт наступної структури і завантажити на Elearn
  - титульний аркуш (приклад наведено вище),
  - текст завдання,
  - скріншоти з назвами використаних в проекті docker образів та docker контейнерів,
  - текст Python коду програм і SQL запитів,
  - скріншоти виконання програм в власному віртуальному середовищі
  - скріншоти всіх таблиць і запитів БД в графічному клієнті
  - посилання на проект на GitHub

Зразок титульного аркушу:

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

**ЛАБОРАТОРНА РОБОТА №9**  
З дисципліні "Кросплатформне програмування Python"

Виконав студент 4-го курсу  
*Микитин Юрій Романович*  
Група ПЗ-22009б,  
**Варіант № 5**

Перевірив:  
к.ф.-м.н. доцент кафедри КН  
*Кириченко Віктор Вікторович*

---

Київ, 2025

# Варіанти завдань

Програмування Python

## Варіант 0

- На платформі Docker, за допомогою файлу `docker-compose.yml`, створити контейнер з СУБД *PostgreSQL* або *MySQL*. Зробити прокидання портів та папок для зберігання БД.
- В цьому контейнері, використовуючи мову Python, створити базу даних. Створити в ній необхідні таблиці з відповідними полями (*предметна область та дані наведені нижче*).
- Визначить типи даних (лічильник, текстовий, числовий тощо) та опис, якщо потрібно.
- Встановіть необхідні властивості полів (розмір поля, маску вводу, значення за замовченням, обмеження та повідомлення про помилку) створених таблиць.
- Визначить первинні та зовнішні ключі в створених таблицях.
- Визначить необхідні зв'язки між таблицями, задайте необхідні параметри забезпечення цілісності даних.
- Заповнить створені таблиці даними ([3 склади](#), [7 клієнтів](#), [17 товарів](#), [22 покупки](#))

**Предметна область:** Магазин одягу (*складається з 4 класів*)

**Сутності та дані:**

**Склади**[Номер складу, адреса, завідувач складом, телефон (маска вводу)],

**Товари**[Код товару, тип (*жіночий, чоловічий, дитячий*), назва одягу, виробник, номер складу, на якому знаходиться, кількість на складі, ціна],

**Клієнти**[Код клієнта, назва клієнта, адреса клієнта, телефон клієнта (маска вводу), контактна особа],

**Продаж**[Код продажу, дата продажу, код клієнта, код товару, кількість купленого одягу, знижка].

### 8. Створіть наступні запити:

Відобразити інформацію по покупкам, яка містить дату покупки, назву товару, назву клієнта, кількість купленого товару, ціну товару. Відсортувати назви клієнтів за алфавітом;

Відобразити увесь одяг за вибраним типом (*запит з параметром*);

Порахувати кількість покупок, яку зробив кожний клієнт (*підсумковий запит*);

Порахувати вартість кожної покупки без урахування та з урахуванням знижки (*запит з обчислювальним полем*);

Порахувати загальну суму грошей, яку витратив кожен клієнт на купівлю одягу (*підсумковий запит*)

Відобразити кількість кожного виду одягу на кожному складі (*перехресний запит*).

### 9. На мові Python написати програму, що підключається до створеної БД, виводить всі таблиці (структур + дані, які в ній зберігаються) та результати виконання запитів в консоль в форматованому вигляді (заголовки стовпців + всі стовпці рівні).

### 10. На платформі Docker створити контейнер з графічним клієнтом (адмінка) для управління БД. Запустити його і підключитись до створеної БД. Переконатись, що всі таблиці і запити створені вірно.

### 11. Завантажити проект на GitHub, попередньо додавши до файлу `.gitignore` всі технічні папки та файли.

### 12. В текстовому редакторі створіть звіт наступної структури і завантажити на Elearn

- титульний аркуш (приклад наведено вище),
- текст завдання,
- скріншоти з назвами використаних в проекті docker образів та docker контейнерів,
- текст Python коду програм і SQL запитів,
- скріншоти виконання програм в власному віртуальному середовищі
- скріншоти всіх таблиць і запитів БД в графічному клієнти
- посилання на проект на GitHub

## Варіант 1

1. На платформі Docker, за допомогою файлу `docker-compose.yml`, створити контейнер з СУБД *PostgreSQL* або *MySQL*. Зробити прокидання портів та папок для зберігання БД.
2. В цьому контейнері, використовуючи мову Python, створити базу даних. Створити в ній необхідні таблиці з відповідними полями (*предметна область та дані наведені нижче*).
3. Визначить типи даних (лічильник, текстовий, числовий тощо) та опис, якщо потрібно.
4. Встановіть необхідні властивості полів (розмір поля, маску вводу, значення за замовченням, обмеження та повідомлення про помилку) створених таблиць.
5. Визначить первинні ключі в створених таблицях.
6. Визначить необхідні зв'язки між таблицями, задайте необхідні параметри забезпечення цілісності даних.
7. Заповнить створені таблиці даними (11 студентів, 3 предмета)

**Предметна область:** Університет (*складається з 3 класів*)

**Сутності та дані:**

**Студенти**[Код студента, Прізвище студента, Ім'я студента, По батькові студента, Адреса студента, телефон студента(маска вводу), курс(1-4, обмеження, повідомлення про помилку), факультет(*аграрного менеджменту, економіки, інформаційних технологій*), група, чи є старостою],

**Предмети**[Код предмету, назва, кількість годин за семестр, кількість семестрів, протягом яких вивчається предмет],

**Складання іспитів**[Код складання, дата складання, код студента, код предмету, отримана оцінка(2-5, обмеження, повідомлення про помилку)].

### 8. Створіть наступні запити:

Відобразить всіх студентів, які є старостами, відсортувати прізвища за алфавітом;  
Порахувати середній бал для кожного студента (*підсумковий запис*);  
Для кожного предмета порахувати загальну кількість годин, протягом яких він вивчається (*запит з обчислювальним полем*);  
Відобразити успішність студентів по обраному предмету (*запит з параметром*);  
Порахувати кількість студентів на кожному факультеті (*підсумковий запис*)  
Відобразити оцінки кожного студента по кожному предмету (*перехресний запис*).

9. На мові Python написати програму, що підключається до створеної БД, виводить всі таблиці (структур + дані, які в ній зберігаються) та результати виконання запитів в консоль в форматованому вигляді (заголовки стовпців + всі стовпці рівні).
10. На платформі Docker створити контейнер з графічним клієнтом (адмінка) для управління БД. Запустити його і підключитись до створеної БД. Переконатись, що всі таблиці і запити створені вірно.
11. Завантажити проект на GitHub, попередньо додавши до файлу `.gitignore` всі технічні папки та файли.
12. В текстовому редакторі створіть звіт наступної структури і завантажити на Elearn
  - титульний аркуш (приклад наведено вище),
  - текст завдання,
  - скріншоти з назвами використаних в проекті docker образів та docker контейнерів,
  - текст Python коду програм і SQL запитів,
  - скріншоти виконання програм в власному віртуальному середовищі
  - скріншоти всіх таблиць і запитів БД в графічному клієнти
  - посилання на проект на GitHub

## Варіант 2

1. На платформі Docker, за допомогою файлу `docker-compose.yml`, створити контейнер з СУБД *PostgreSQL* або *MySQL*. Зробити прокидання портів та папок для зберігання БД.
2. В цьому контейнері, використовуючи мову Python, створити базу даних. Створити в ній необхідні таблиці з відповідними полями (*предметна область та дані наведені нижче*).
3. Визначить типи даних (лічильник, текстовий, числовий тощо) та опис, якщо потрібно.
4. Встановіть необхідні властивості полів (розмір поля, маску вводу, значення за замовченням, обмеження та повідомлення про помилку) створених таблиць.
5. Визначить первинні ключі в створених таблицях.
6. Визначить необхідні зв'язки між таблицями, задайте необхідні параметри забезпечення цілісності даних.
7. Заповнить створені таблиці даними ([14 книг, 9 читачів, 11 видач](#))

**Предметна область:** Бібліотека інституту (*складається з 3 класів*)

**Сутності та дані:**

**Книги**[інвентарний номер книги, автор книги, назва, розділ (*технічна, художня, економічна*), рік видання, кількість сторінок, ціна, вид (*посібник, книга, періодичне видання*), кількість примірників, максимальний термін нахождення у читача (*кількість днів*)].

**Читачі**[номер читацького квитка, прізвище читача, ім'я читача, телефон читача(маска вводу), адреса, курс(1-4, обмеження, повідомлення про помилку), група],

**Видача книжок**[Код видачі, дата видачі, номер квітка читача, номер книги].

### 8. Створіть наступні запити:

Відобразити всі книги, які були видані після 2001 року. Відсортувати назви за алфавітом;  
Порахувати кількість книг кожного виду (*підсумковий запит*);  
Відобразити всіх читачів, які брали посібники в бібліотеці. Відсортувати прізвища за алфавітом;  
Відобразити всі книги за указаним розділом (*запит з параметром*);  
Дляожної книги, яка була видана читачу, порахувати кінцевий термін її повернення в бібліотеку (*запит з обчислювальним полем*);  
Порахувати кількість посібників, книг та періодичних видань в кожному розділі (*перехресний запит*)

9. На мові Python написати програму, що підключається до створеної БД, виводить всі таблиці (структур + дані, які в ній зберігаються) та результати виконання запитів в консоль в форматованому вигляді (заголовки стовпців + всі стовпці рівні).
10. На платформі Docker створити контейнер з графічним клієнтом (адмінка) для управління БД. Запустити його і підключитись до створеної БД. Переконатись, що всі таблиці і запити створені вірно.
11. Завантажити проект на GitHub, попередньо додавши до файлу `.gitignore` всі технічні папки та файли.
12. В текстовому редакторі створіть звіт наступної структури і завантажити на Elearn
  - титульний аркуш (приклад наведено вище),
  - текст завдання,
  - скріншоти з назвами використаних в проекті docker образів та docker контейнерів,
  - текст Python коду програм і SQL запитів,
  - скріншоти виконання програм в власному віртуальному середовищі
  - скріншоти всіх таблиць і запитів БД в графічному клієнти
  - посилання на проект на GitHub

## Варіант 3

1. На платформі Docker, за допомогою файлу `docker-compose.yml`, створити контейнер з СУБД *PostgreSQL* або *MySQL*. Зробити прокидання портів та папок для зберігання БД.
2. В цьому контейнері, використовуючи мову Python, створити базу даних. Створити в ній необхідні таблиці з відповідними полями (*предметна область та дані наведені нижче*)
3. Визначить типи даних (лічильник, текстовий, числовий тощо) та опис, якщо потрібно.
4. Встановіть необхідні властивості полів (розмір поля, маску вводу, значення за замовченням, обмеження та повідомлення про помилку) створених таблиць.
5. Визначить первинні ключі в створених таблицях.
6. Визначить необхідні зв'язки між таблицями, задайте необхідні параметри забезпечення цілісності даних.
7. Заповнить створені таблиці даними (**17 працівників, 8 проектів, відділи: програмування, дизайн, інформаційних технологій, посади: інженер, редактор, програміст**)

**Предметна область:** Відділ кадрів (*складається з 5 класів*).

**Сутності та дані:**

**Співробітники**[Код співробітника, прізвище співробітника, ім'я, по батькові, адреса, телефон(маска вводу), освіта (*спеціальна, середня, вища*), код відділу, код посади],

**Відділи**[Код відділу, назва відділу, телефон (маска вводу), № кімнати (обмеження 701-710, повідомлення про помилку)],

**Посади**[Код посади, Посада, оклад, премія(відсоток від окладу)],

**Проекти**[Номер проекту, назва проекту, термін виконання, розмір фінансування],

**Виконання проектів**[Код виконання, номер проекту, код відділу в якому виконується проект, дата початку].

### 8. Створіть наступні запити:

Відобразити всіх робітників, які мають оклад більший за 2000 грн. Відсортувати прізвища за алфавітом;

Порахувати середню зарплатню в кожному відділі (*підсумковий запит*);

Відобразити всі проекти, які виконуються в обраному відділі (*запит з параметром*);

Порахувати кількість працівників у кожному відділі (*підсумковий запит*);

Порахувати розмір премії для кожного співробітника (*запит з обчислювальним полем*);

Порахувати кількість робітників які мають *спеціальну, середню, вищу* освіту у кожному відділі (*перехресний запит*).

### 9. На мові Python написати програму, що підключається до створеної БД, виводить всі таблиці (структур + дані, які в ній зберігаються) та результати виконання запитів в консоль в форматованому вигляді (заголовки стовпців + всі стовпці рівні).

### 10. На платформі Docker створити контейнер з графічним клієнтом (адмінка) для управління БД. Запустити його і підключитись до створеної БД. Переконатись, що всі таблиці і запити створені вірно.

### 11. Завантажити проект на GitHub, попередньо додавши до файлу `.gitignore` всі технічні папки та файли.

### 12. В текстовому редакторі створіть звіт наступної структури і завантажити на Elearn

- титульний аркуш (приклад наведено вище),
- текст завдання,
- скріншоти з назвами використаних в проекті docker образів та docker контейнерів,
- текст Python коду програм і SQL запитів,
- скріншоти виконання програм в власному віртуальному середовищі
- скріншоти всіх таблиць і запитів БД в графічному клієнті
- посилання на проект на GitHub