

PYTHON+SQL:

ЯК ПОЧАТИ ВИКОРИСТОВУВАТИ БД І ПИСАТИ SQL-ЗАПИТИ

Частина 1

Навчальний центр
інформаційних технологій

cbs.com.ua

CyberBionic
systematics

Відео курси
з програмування

itvdn.com

 **ITVVDN**
IT VIDEO DEVELOPERS NETWORK

Introduction



Костянтин Зівенко

python backend developer in waylight.me
тренер курсу «python developer» в [CBS](https://CBS.com)

[!\[\]\(339a16584d5da0f0a3ca4e9ec17bf6a1_img.jpg\) посилання на профіль тут](#)

[!\[\]\(a870788d6ed9b8fd294b7654a8c8526b_img.jpg\) посилання на профіль тут](#)

Я – класичний приклад «світчера» (від англ. switch). Приблизно 4,5 роки тому вирішив повністю змінити напрямок діяльності і місце життя. На той час я не мав досвіду написання коду, англійської мови, спеціальних знань.

У мене все вийшло)

Сьогодні – я працюю в ІТ в якості розробника і ділюсь знаннями з тими, хто хоче пройти цей шлях так само, як пройшов його я.

Python+SQL: як почати використовувати БД і писати SQL-запити

План

День 1. Підготовка та перші кроки.

1. Про вебінар. Цілі, завдання, очікуваний результат.
Як працюємо.
2. Введення в реляційні БД.
3. Коротко – що таке SQL
4. SQLite + Python
 1. Проектуємо невелику БД: UML, нормалізація, відношення у таблицях – коротко. Типи даних SQLite. Бібліотека sqlite3 – основи використання.
 2. Створюємо БД: оператори CREATE TABLE, INSERT INTO. Як створити фейкові дані – бібліотека Faker.
5. Клієнт для керування БД: встановлюємо та налаштовуємо Dbeaver.

День 2. Робота з БД – основні SQL-запити.

1. Прості запити з SELECT (DISTINCT, COUNT, WHERE, AND, OR BETWEEN, IN\NOT IN, ORDER BY, LIKE, MIN\MAX\AVG, UNION\INTERSECT\EXCEPT)
2. JOIN ()
3. Підзапити (WHERE EXIST, ANY\ALL квантіфікатори)
4. Що далі?
 1. Ви отримали БД, базові навички і приклади роботи. Не зупиняйтесь!
 2. На що вистачає цих знань студентам, які проходять наш курс.

Python+SQL: як почати використовувати БД і писати SQL-запити

Про вебінар: мета, завдання, очікуваний результат. Як працюємо

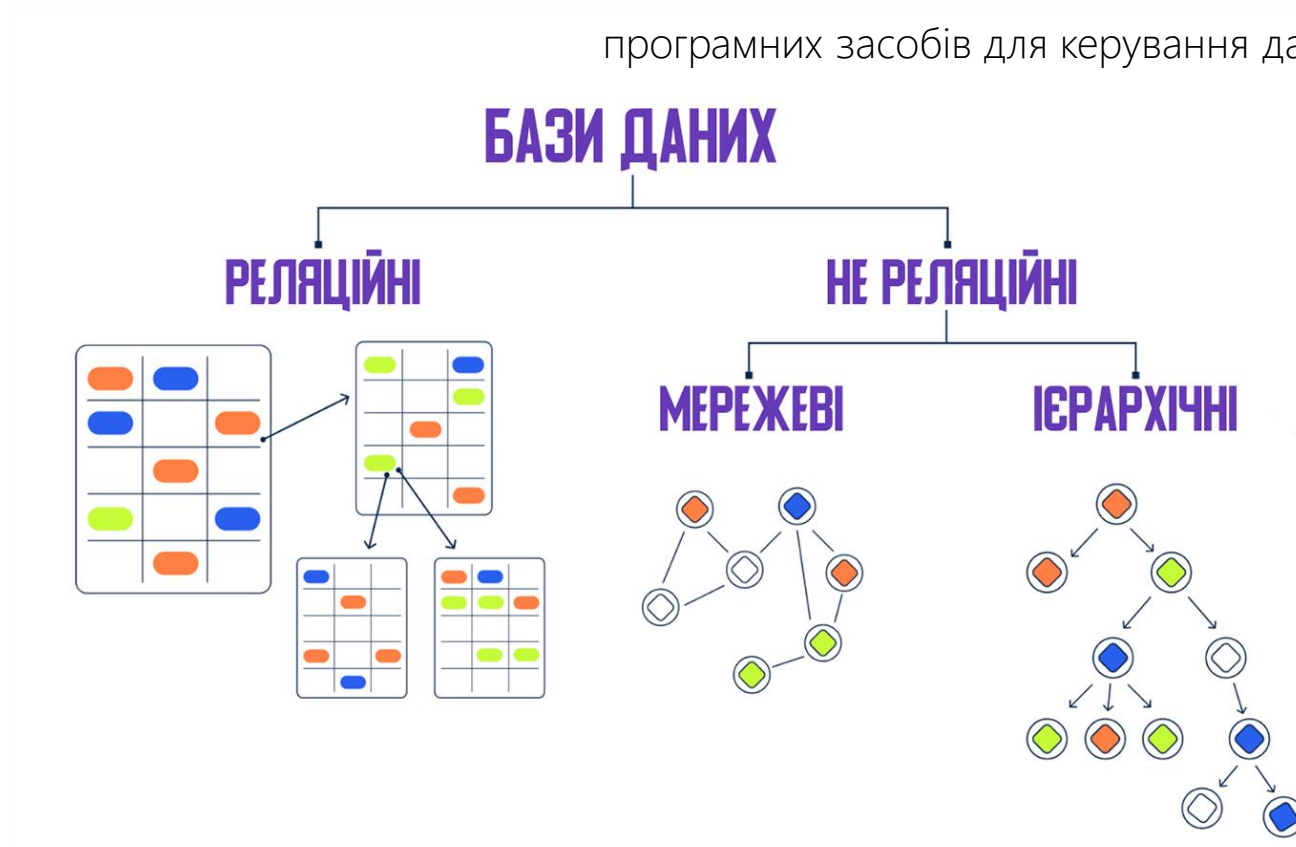
1. Чому саме ця тематика?
2. Що очікувати від вебінару слухачам?
3. Як ми працюємо
 1. Порядок подачі матеріалу (Презентація. Матеріали в репозиторії. .ipnb файл як конспект лекцій. Як працювати з GIT і GitLab – [тиць тут](#))
 2. Порядок відповіді на виникаючі питання – чому я буду мінімально відповідати на питання в чаті і буду намагатись максимально відповідати на питання в коментарях під відео (створюємо можливість для тих – хто не може дивитись в онлайн, тримаємо графік) . Перед тим як писати питання – прочитайте написані раніше, можливо там вже є відповідь.
 3. Посилання на репозиторій де будуть ВСІ матеріали вебінару – будуть закріплені під відео.
 4. За результатами вебінару ви отримаєте:
 1. Презентацію з матеріалами
 2. БД з якою зможете експериментувати і тренуватись.
 3. Інтерактивний конспект лекцій – де написаний код для всіх кроків докладно пояснено і може бути запущений вами.
 4. Мої відповіді на питання написані під відео.

Введення в реляційні бази даних

База даних (БД, DataBase) – це набір взаємно пов'язаних даних.

СУБД – відповідає за підтримку мови БД, організацію створення БД, внесення даних, збереження, маніпулювання даними, оптимізацію процесів роботи з даними й т. ін.

Система управління базами даних (СУБД, СКБ, англ. Database Management System, DBMS) – комплекс програмних засобів для керування даними.



Python+SQL: як почати використовувати БД і писати SQL-запити

1. ІЄРАРХІЧНА МОДЕЛЬ БАЗИ ДАНИХ

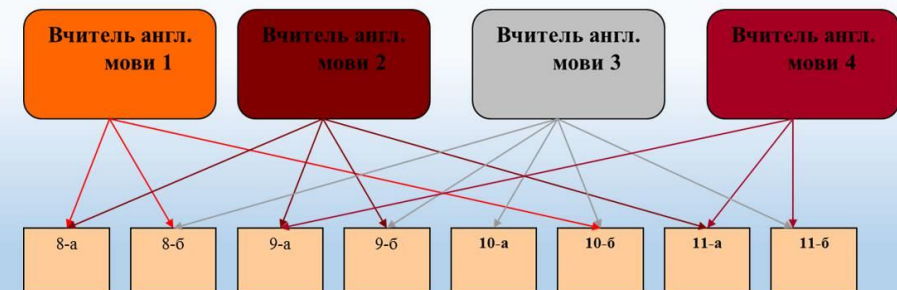
Вона являє собою структуру даних, які впорядковані за підляганням від загального до конкретного, що нагадує «дерево» (граф), тому має такі складові: рівень, вузол, зв'язок.



Не реляційні СУБД з'явилися дуже давно – і з деякими ви напевне знайомі. Приклад ієрархічної організації даних – каталог звичайної бібліотеки (докладніше – [тиць тут](#))

Мережева модель даних – розвиток ієрархічної моделі. Відмінність – вузол може бути нащадком не одного предка, а будь-скільких. Це не каша, а досить складна і суворая математична теорія, до-речі ☺ (докладніше – [тиць тут](#))

МЕРЕЖЕВА МОДЕЛЬ



Python+SQL: як почати використовувати БД і писати SQL-запити

Але сьогодні ми розмовляємо про [реляційні бази даних](#).

Теоретичною базою реляційних СУБД є [реляційна алгебра](#).

Досить проста презентація, яка демонструє базові інструменти – [тут](#).

Лекція про множини і операції з ними в Python, яка нагадає Вам роботу дуже схожих операцій – доступна на каналі

[CodeUA](#) [тут](#). Дивіться останню чверть лекції.

Операція об'єднання

$$R(L) = R1(L) \cup R2(L) = \{r | r \in R1 \vee r \in R2\}$$

R1		R2		R1 U R2	
A	B	A	B	A	B
a1	b1	a1	b1	a1	b1
a1	b2	a2	b1	a1	b2
a2	b3			a2	b1
				a2	b3

Операції реляційної алгебри

Операція перетину

$$R(L) = R1(L) \cap R2(L) = \{r | r \in R1 \& r \in R2\}$$

R1		R2		R1 ∩ R2	
A	B	A	B	A	B
a1	b1	a1	b1	a1	b1
a1	b2	a2	b1		
a2	b3				

Різниця

Різниця відношень $R1 \setminus R2 : R = R1 - R2$.
Різницею $(R1 - R2)$ називається множина кортежів, що належать $R1$, але не приналежних $R2$. Вимоги до арності - такі ж.

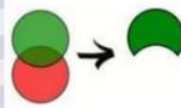
Персони

Имя	Возраст	Вес
Harry	34	80
Donald	29	70
Helena	54	54
Peter	34	80

Персонажі

Имя	Возраст	Вес
Daffy	24	19
Donald	29	70
Scrooge	81	27

Имя	Возраст	Вес
Harry	34	80
Helena	54	54
Peter	34	80
Daffy	24	19
Donald	25	23
Scrooge	81	27



Обозначение	Определение	LEAP
$R_1 - R_2$	$\{r : r \in R_1 \wedge r \notin R_2\}$	$r = (R1) \text{ difference } (R2)$ $r = (R1) \text{ minus } (R2)$

Пример:



$$R[Q.T] - S = \begin{bmatrix} 5 & a \\ 3 & a \\ 9 & a \\ 2 & b \\ 4 & b \end{bmatrix} \cap \begin{bmatrix} 5 & a \\ 10 & b \\ 15 & c \\ 2 & d \\ 6 & a \\ 1 & b \end{bmatrix} = \begin{bmatrix} 3 & a \\ 2 & b \\ 4 & b \end{bmatrix}$$

Python+SQL: як почати використовувати БД і писати SQL-запити

Реляційна алгебра визначає комплекс операцій з відношеннями (таблицями): поєднання, різницю, перетин, додток ...

Всі ці операції можуть бути виражені через мову програмування SQL, що ми і будемо з Вами вчитись робити.

Термінологія:

- **Сутність** – те, що ми описуємо (якийсь конкретний товар, якась конкретна людина, замовлення і т.ін.)
- **Таблиця** (відношення) – це множина якихось сутностей
- Стовпчик таблиці – **атрибути** – один з параметрів описуваної сутності
- Рядок таблиці (**запис**, кортеж) – набір атрибутів з конкретними значеннями
- **Фінальний набір** – результат SQL запиту. Це завжди таблиця (навіть коли результат – одне просте число)

Таблиці (стовпці – атрибути, рядки – сутності (дані), шапка таблиці – імена атрибутів)

Коротко – що таке SQL

SQL – спеціалізована непроцедурна мова програмування.

SQL стандартизований. Діючий на сьогодні стандарт – ANSI SQL-92. Як правило, різні БД підтримують цей стандарт і додають свої розширення (свій діалект).

Складається з декількох груп операторів:

- DDL (Data Definition Language) – CREATE, DROP, ALTER
- DML (Data Manipulation Language) – SELECT, INSERT, UPDATE, DELETE
- TCL (Transaction Control Language) – COMMIT, ROLLBACK, SAVEPOINT
- DCL (Data Control Language) – GRANT, REVOKE, DENY

Python+SQL: як почати використовувати БД і писати SQL-запити

Світ SQL баз даних – досить широкий і неоднорідний

За архітектурою побудови:

- Вбудовані ([SQLite](#), ...)
- Клієнт-серверні (PostgreSQL, MySQL, ...)
- Файл-серверні (Microsoft Access, ...)



Приклади популярних клієнт-серверних СУБД:

- MySQL
- PostgreSQL
- Oracle
- MS SQL
- MariaDB

A screenshot of the SQLite website. The header includes the SQLite logo and a navigation bar with links: Home, About, Documentation, Download, License, Support, Purchase. Below the navigation bar is the text "Most Widely Deployed and Used Database Engine". The main content area states "SQLite is likely used more than all other database engines combined. Billions and billions of copies of SQLite exist in the wild. SQLite is found in:" followed by a bulleted list of applications and devices. At the bottom, it mentions "Since SQLite is used extensively in every smartphone, and there are more than 4.0 billion (4.0e9) smartphones in active use, each holding hundreds of SQLite database files, it is seems likely that there are over one trillion (1e12) SQLite databases in active use." The footer of the website shows "Small. Fast. Reliable. Choose any three." and "Слайд: 1 из 8".

Small. Fast. Reliable.
Choose any three.

Слайд: 1 из 8

Home About Documentation Download License Support Purchase

Most Widely Deployed and Used Database Engine

SQLite is likely used more than all other database engines combined. Billions and billions of copies of SQLite exist in the wild. SQLite is found in:

- Every Android device
- Every iPhone and iOS device
- Every Mac
- Every Windows10 machine
- Every Firefox, Chrome, and Safari web browser
- Every instance of Skype
- Every instance of iTunes
- Every Dropbox client
- Every TurboTax and QuickBooks
- PHP and Python
- Most television sets and set-top cable boxes
- Most automotive multimedia systems
- Countless millions of other applications

Since SQLite is used extensively in every smartphone, and there are more than 4.0 billion (4.0e9) smartphones in active use, each holding hundreds of SQLite database files, it is seems likely that there are over one trillion (1e12) SQLite databases in active use.

Python + SQLite



Передумови для роботи:

Далі ми, нарешті, перейдемо до написання коду)

Ми будемо це робити у [блокноті jupyter](#)

Це зручно для лекцій і презентацій і вам у репозиторії буде доступний цей блокнот – сформований як конспект лекції – з усіма текстовими поясненнями і кодом, який ви можете запустити.

Я буду робити це використовуючи IDE PyCharm Professional Edition, це доступно, також, у Visual Studio Code.

Якщо ви не хочете або не можете встановлювати IDE, ви можете використовувати cloud-середовище від Google [Colab](#)

В будь-якому разі, для отримання повної функціональності від цього конспекту Вам необхідно встановити залежності – які описані у файлі requirements.txt.

Python+SQL: як почати використовувати БД і писати SQL-запити

Яку БД ми сьогодні робимо:

Ми створимо невелику БД, як досить часто роблять в прикладах – програми соціальної мережи. Ми маємо користувачів (users) із своїми атрибутами. Користувачі можуть писати пости (posts), які також мають свої атрибути, та можуть писати коментарі до постів (comments). Користувачі, також, можуть ставити лайки постам, які їм сподобались (likes). У користувачів є атрибут, якого може бути декілька варіантів – emails. Так як їх може бути декілька – ми винесли їх в окрему таблицю.

Python+SQL: як почати використовувати БД і писати SQL-запити

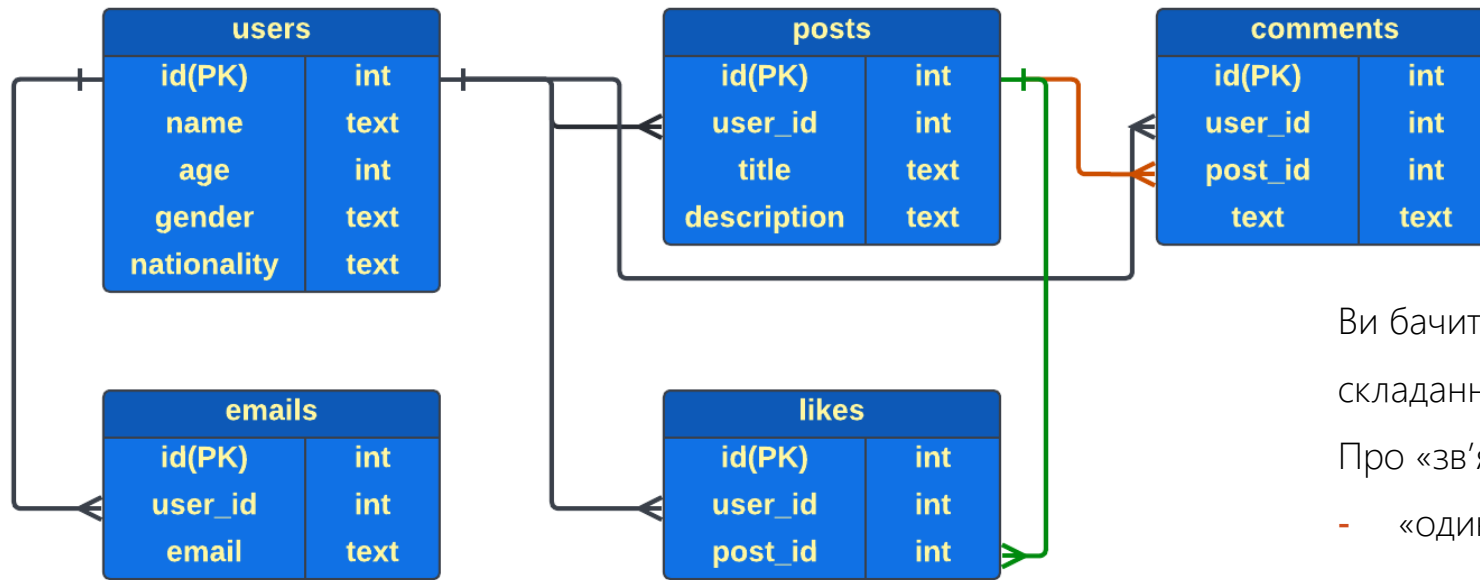
Я сьогодні буду оминати таку тему як «нормалізація бази даних» - в межах короткого курсу це досить важко викласти. Розумійте нормалізацію як процес розділення складних таблиць на більш прості таблиці з утворенням зв'язків між простими таблицями. Такі дії дозволяють СУБД контролювати цілісність даних і уникати колізій. А також мати дані у більш компактному форматі. Це робиться на базі абсолютно чітких математичних правил і визначень – але досить зрозумілих і логічних, якщо вникнути.

Докладніше про це математичною мовою [тут](#).

Докладніше про це простішою мовою – [тут](#).

Я дуже поверхнево торкнусь сьогодні деяких понять – цілісність даних, зв'язки між таблицями – що дасть Вам розуміння про що це взагалі. Ми будемо говорити про це на практиці – створюючи нашу БД.

Python+SQL: як почати використовувати БД і писати SQL-запити



ER-діаграма нашої БД:

Ви бачите ER-діаграму нашої БД. Докладніше про правила складання таких діаграм – [тут](#).

Про «зв'язки між таблицями». Існує декілька:

- «один до одного»
- «один до багатьох»
- «багато до багатьох»

Розберемо зв'язки в нашій БД:

Таблиці users і posts матимуть відношення «один до багатьох» - так як один користувач може написати багато публікацій. Подібним чином один користувач може зробити багато коментарів, а одна публікація може мати декілька коментарів. Тобто таблиці users і posts матимуть зв'язки «один-до-багатьох» із таблицею «comments». Така ж ситуація з likes таблицею. Окрім цього ми зауважили на початку, що один користувач може мати декілька e-mail, тобто зв'язок між users та emails – також «один до багатьох».

Python+SQL: як почати використовувати БД і писати SQL-запити

Працюємо з SQLite (модуль sqlite3, intro in SQL and SQLoie.ipynb – в репозиторії)

- Створюємо БД і підключаємось
- Створюємо таблиці, обмеження і коротко про цілісність даних
- Наповнюємо таблицю даними, модуль Faker



Dbeaver - це клієнтська програма для управління базами даних (БД). Воно використовує програмний інтерфейс JDBC для взаємодії з реляційними базами даних через драйвер JDBC

Ми будемо його використовувати і взагалі – це зручний інструмент для роботи з різноманітними базами даних, який (або подібний) необхідно мати під рукою.

[Скачати можна тут.](#)

Домашнє завдання після першого дня:

- Відтворити у себе локально весь процес з jupyter ноутбука – intr0_in_SQL_and_SQLite.ipynb.

Можливий варіант дій:

- Клонувати собі [репозиторій](#)
- Встановити залежності
- Пройти по конспекту
- Встановити Dbeaver і підключитись до створеної БД.
- Спробувати написати до створених таблиць запит `SELECT * FROM <table_name>;`

Q&A

Онлайн курс з тренером Python Developer з нуля і до рівня спеціаліста

До початку занять залишилося:

11 : 7 : 3 : 10

Дні

Години

Хвилини

Секунди

Старт курсу:

■ 12 грудня, 09:00-11:00 (GMT+2),
Українська

■ 12 грудня, 19:00-21:00 (GMT+2),
Українська

■ 12 грудня, 09:00-11:00 (GMT+2),
Міжнародна

■ 12 грудня, 19:00-21:00 (GMT+2),
Міжнародна

ЗАПИСАТИСЯ НА КУРС

ЗАПИСАТИСЯ НА ПРОБНИЙ УРОК



Формат навчання
LIVE ONLINE



Графік:
2 РАЗИ НА ТИЖДЕНЬ



Тривалість:
5 МІС. / 88 ГОДИН



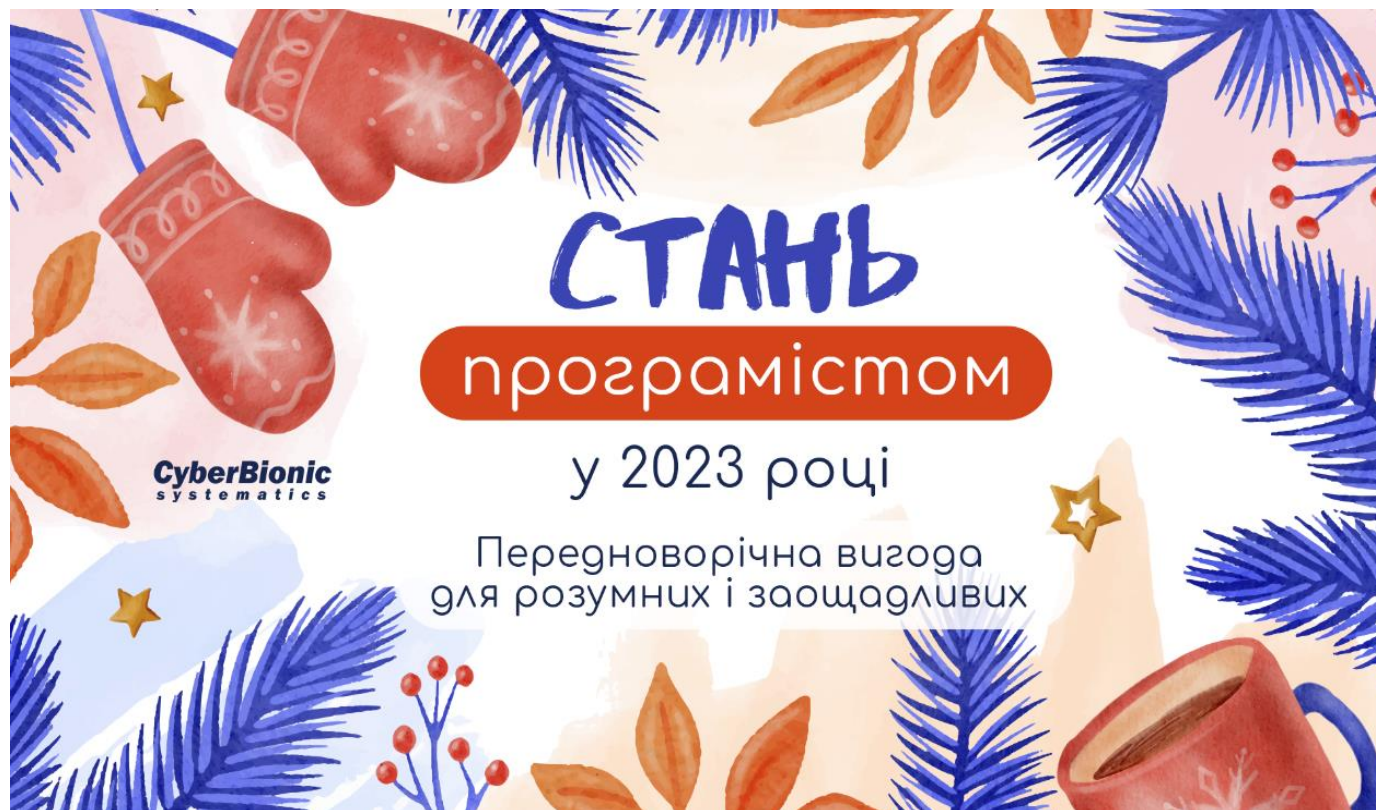
В одній групі:
ВІД 6 ДО 25 ОСІБ



Під час та після навчання
HR-ПІДТРИМКА



Знижка 30% на курс Python Developer



До 15 грудня пропонуємо економію у **6405 грн** на спеціальність! Можна оплатити частинами і почати навчання в грудні чи січні!

В групі до 25 місць, тож пропозиція обмежена.

[Реєструйтесь зараз!](#)

Додаткові ресурси для студентів

[ITVDN](#) – відео курси з програмування, в тому числі комплексна програма спеціальності Python Developer

ВІДЕО КУРСИ ЗА СПЕЦІАЛЬНІСТЮ PYTHON DEVELOPER

● Python 3.8.6 Стартовый	🕒 4 год 43 хв
● Python 3.8.6 Базовый	🕒 5 год 52 хв
● Python Углубленный	🕒 13 год 51 хв
● Flask	🕒 6 год 28 хв
● Создание проекта на Python и Flask	🕒 11 год 12 хв
● Основы MongoDB	🕒 2 год 57 хв
● PostgreSQL	🕒 3 год 5 хв
● MySQL Базовый	🕒 7 год 58 хв
● HTML5 & CSS3 Стартовый	🕒 5 год 11 хв
● JavaScript Стартовый	🕒 8 год 41 хв
● Django Стартовый	🕒 18 год 27 хв
● Django Базовый	🕒 13 год 10 хв
● Основы работы с Git	🕒 2 год 13 хв
● Методологии управления проектами. Вступление в SCRUM	🕒 1 год 53 хв
● Подготовка к собеседованию в IT компании. Вопросы и ответы. Хитрости. Трюки.	🕒 1 год 50 хв

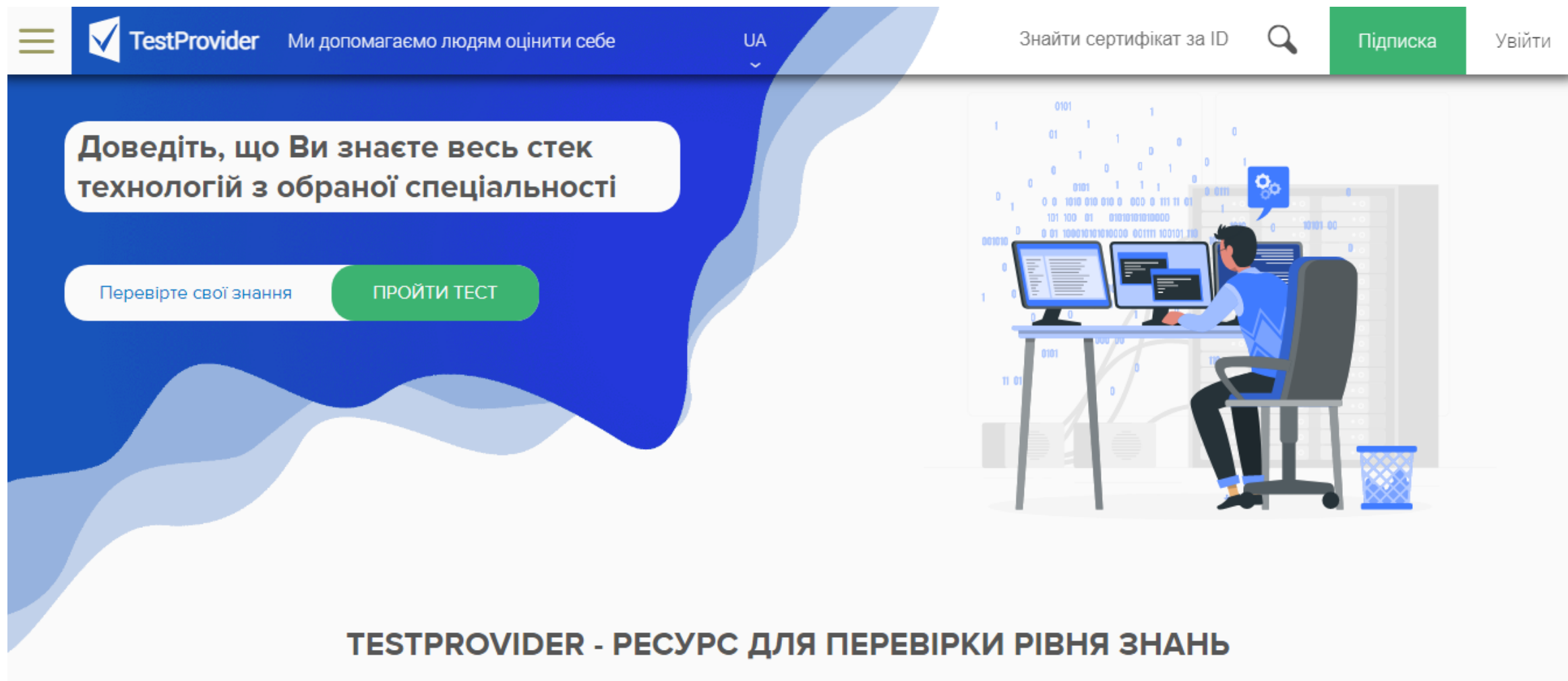
ЗАПИСИ БЕЗКОШТОВНИХ ВЕБІНАРІВ

● Как стать Python разработчиком в 2021 году	1 год 53 хв
● Создание простой казуальной игры на Python.	1 год 51 хв
● Создание чат-бота "прогноз погоды" на Python.	2 год 54 хв
● Объектно-ориентированное программирование в Python.	2 год 2 хв
● Автоматизация парсинга сайтов на Python.	2 год 4 хв
● Создаем нейронную сеть на Keras с нуля.	2 год 14 хв
● Структуры данных в Python. Уровень Advanced.	2 год 3 хв
● Знакомство с Python - видеоуроки ITVDN	2 год 13 хв
● Как стать Python разработчиком?	1 год 31 хв
● Создание первого веб-сайта при помощи Python и Django - видео уроки ITVDN	1 год 28 хв
● Создание социальной сети с помощью Python	1 год 46 хв
● Размещение веб-приложений на Python в облаке Microsoft Azure	2 год 4 хв
● Архитектура Flask приложений.	2 год 4 хв
● SQLAlchemy ORM: удобная работа с базами данных на Python - видеоуроки ITVDN	1 год 36 хв
● PySharp - практическое использование Python в .NET приложениях	2 год 40 хв
● Что нового в Python? Спецификация PEP 8.	1 год 40 хв

● Обязательные курсы ● Дополнительные курсы ● Бесплатные вебинары

Додаткові ресурси для студентів

[Testprovider](#) – сервіс онлайн тестування з Python, MySQL та іншим технологіям з курсу підготовки Python Developer



The image shows the TestProvider website interface. The header is blue with the TestProvider logo, a tagline, a language selector (UA), a search bar for certificates, and buttons for 'Підписка' (Subscription) and 'Увійти' (Login). The main content area has a blue background with a white box containing the text 'Доведіть, що Ви знаєте весь стек технологій з обраної спеціальності' (Prove that you know the whole stack of technologies from the chosen specialty). Below this is a button 'ПРОЙТИ ТЕСТ' (Take Test) and a link 'Перевірте свої знання' (Check your knowledge). On the right, there is an illustration of a person sitting at a desk with two monitors, working on a computer, with binary code and a gear icon in the background.

TESTPROVIDER - РЕСУРС ДЛЯ ПЕРЕВІРКИ РІВНЯ ЗНАНЬ

День 2. Робота з БД – основні SQL-запити:

- Повна вибірка: `SELECT * FROM <table_name>;`
- Вибірка конкретних атрибутів: `SELECT <atr_1>, <atr_2>, ..., <atr_n> FROM <table_name>;`
- Прості операції (+ - * /). [Вбудовані функції.](#)
- DISTINCT
- COUNT
- WHERE
- AND, OR
- BETWEEN
- IN, NOT IN
- ORDER BY
- MIN, MAX, AVG
- LIKE
- GROUP BY
- JOIN
 - INNER JOIN
 - LEFT JOIN, RIGHT JOIN
 - FULL JOIN

Основна частина матеріалу і прикладів для цієї частини вебінару має прикладний характер і максимально ефективність досягається при практичному виконання разом з викладачем, а потім самотійно. Тому весь цей матеріал зосереджений у конспекті лекції – `intro_in_SQL_and_SQLite.ipynb`

Запрошую перейти)

Інформаційний відеосервіс для розробників програмного забезпечення

Навчальний центр
інформаційних технологій
cbs.com.ua

CyberBionic
systematics

Відео курси
з програмування
itvdn.com



CyberBionic
systematics

Microsoft Partner



 **Windows Azure**

