МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

Факультет радіоелектроніки, комп'ютерних систем та інфокомунікацій

**Лабораторна робота № 5 `**

з «Обробка даних засобами Python»

(назва дисципліни)

тема: «Робота з базами даних SQL в Python»

Виконав: студент 5 курсу групи № 555вМн

напряму підготовки (спеціальності)

123 Системне програмування

(шифр і назва напряму підготовки (спеціальності))

Логачов М.Г.

(прізвище й ініціали студента)

Прийняв: д.т.н., доцент

Дергачов К.Ю.

(посада, науковий ступінь, прізвище й ініціали)

Національна шкала: \_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

Харків – 2023

Лабораторна робота № 5

**Тема**: Робота з базами даних SQL в Python.

**Мета роботи :** Дослідження методів роботи з базами даних у Python.

**Хід роботи**

1. Вивчити методи роботи з базами даних SQLLite у Python.
2. Відповідно до заданої предметної області створити базу даних що складається не менше ніж з (!) 4-х взаємопов'язаних таблиць. (!) Кожна таблиця повинна містити щонайменше 4 полів та 15 записів.
3. Розробити та виконати 5 SQL-запитів до бази даних на вибірку даних різних типів (простий, перехресний, з використанням різних таблиць, визначення агрегованих характеристик тощо)

Варіант №8

Предметна галузь : Пункт прокату обладнання

Згідно отриманого варіанту побудована графічна схема моделі бази даних для пункту прокату гірськолижного обладнання, яка зображена на рис.1.

Рисунок 1. Схематичне зображення моделі бази дани

Graphical user interface, diagram

Description automatically generated

Таблиця *Clients* містить персональну інформацію про кліента.

* *client\_id –* індивідуальний номер, який присвоюється автоматично
* *client\_name –* ім’я користувача
* *contact\_phone* – номер телефону
* *sex –* стать кліента

Таблиця *Items* містить інформацію про інвентар.

* *item\_id –* індsвідуальний номер, який присвоюється автоматично
* *item\_type –* назва обладнення
* *item\_price* – вартість оренди за добу
* *status –* доступність обладнання для оренди

Таблиця *Staff* містить інформацію про персонал сервісу.

* *worker\_id –* індивідуальний номер, який присвоюється автоматично
* *worker\_name –* назва обладнення
* *department* – вартість оренди за добу
* *job\_title –* доступність обладнання для оренди

Таблиця *Rental Info* містить інформацію про оренду.

* *rental\_id –* індивідуальний номер, який присвоюється автоматично
* *client\_id –* індивідуальний номер клієнта (зовнішній ключ який через відношення одним до багатьох зв’язуеться з таблицею *Clients* )
* *item\_id –* індивідуальний номер обладнення (зовнішній ключ який через відношення одним до багатьох зв’язуеться з таблицею *Items*)
* *assigned\_worker\_id* – співробітник, який видав інвентар (зовнішній ключ який через відношення одним до багатьох зв’язуеться з таблицею *Staff*)
* *start\_date –* дата початку оренди
* *expiration\_date –* дата закінчення оренди
* *total\_price* – загальна вартість оренди (обчислюється як добуток вартості орендованого обладння на добу та кількісь днів оренди)
* *status –* статус оренди

Код програми та приклади SQL представлені за посиланням на [GitHub](https://github.com/lohachovKhai/geodata_processing/tree/main/Lab_5)

Вхідні дані, для простоти імплементації, представлені у вигляді масивів даних.

Дані до таблиці *Clients*

Text

Description automatically generated

Дані до таблиці *Staff*

Text

Description automatically generated with medium confidence

Дані до таблиці *Items*

Text

Description automatically generated

Дані до таблиці *Rental Info*

Text

Description automatically generated with medium confidence

За допомогою бібліотеки SQLite3 та мови програмування Python - для створення таблиць виконаємо SQL запити:

Text

Description automatically generated

Виконаємо CRUD операції над вхідними даними.

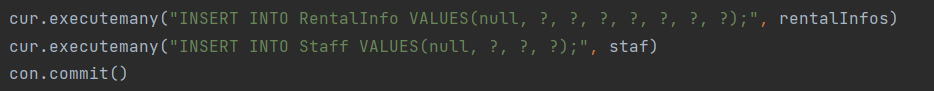
* C - *CREATE*
* R - *READ*
* U - *UPDATE*
* D - *DELETE*

Створимо запис у таблицях *Clients* та *Equipment*Text

Description automatically generated

Text

Description automatically generatedПеред тим, як створити запис у теблиці *Rental Info* та *Staff,* необхідно обчислити загальну вартість оренди відповідного обладнання. Для цього скористаємось функцією *totalPriceCounter(). Rental Info* пов’зана з таблицею *Equipment*  по зовнішньому ключу *item\_id .* Скористаємось цим, та отримаємо вартість оренди за добу виконавши запит *SELECT.*

Отримані результати заносимо до відповідних таблиць

Для перегляду даних, які були занесені до таблиць можна скористатись SQL запитом **("SELECT \* FROM {TABLE\_NAME} ;")** або за допомогою графічного інтерфейсу програми DB Browser (SQLite), результати відображення якої представлені на (рис.1 – 4).

Рисунок 2. Таблиця *Clients* зі значеннями

Table

Description automatically generated

Рисунок 3. Таблиця *Equipment* зі значеннями

*Table

Description automatically generated*

Рисунок 4. Таблиця *Staff* зі значеннямиTable

Description automatically generated

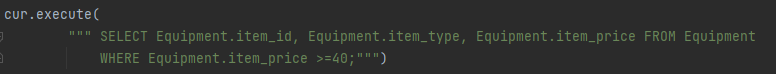
Рисунок 5. Таблиця *Rental Info* зі значеннями

Table

Description automatically generated

Приклади запитів та отримані результат

1. Виконаємо операцію *READ* та отримаємо перелік обладнання, вартість оренди на добу котрого, становить більше 40 грошових єдиниць.

****SQL запит:

Text

Description automatically generatedОтриманий результат

1. Виконаємо операцію *UPDATE* для користувача ***Mona Kulas***, змінивши йому номер телефону.

Text

Description automatically generatedДля кращого відображення результату, спочтку зробимо запит до поточної таблиці *Clients* та виведемо результат *(BEFORE UPDATE EXECUTION)* .

Graphical user interface, text

Description automatically generatedНаступним кроком виконаємо SQL запит UPDATE та відобразимо зміни, виконавши SELECT

Text

Description automatically generatedText

Description automatically generated

1. Виконаємо операцію *DELETE* для таблиці зі співробітниками.

Для цього створемо нового співробітника *Jackson Astraia*, а потім видалимо його з таблиці, імітуючи звільнення.

Graphical user interface

Description automatically generatedЯк і в пунткті 2, відобразимо «До» та «Після» виконання операції DELETE

Text

Description automatically generatedТаблиця *Staff* до створення нового співробітника

Text

Description automatically generatedЗапит на створення нового співробітника

Text

Description automatically generatedТаблиця *Staff*  після створення нового співробітника

SQL запит для видалення елементу з таблиці Graphical user interface, text

Description automatically generated

Таблиця *Staff*  після створення нового співробітника

Text

Description automatically generated

1. Детальне відображення інформації щодо орендованого обладнання

Text

Description automatically generatedSQL запит:

Text

Description automatically generatedОтриманий результат

1. Пошук користувача, у якого скоро сплине строк оренди

SQL запитText

Description automatically generated:

Отриманий результат:

1. Запит на пошук найпопулярнішого товару (тобто товару, який найчастіше був орендований)

Text

Description automatically generatedSQL запит

Text

Description automatically generatedОтриманий результат

Висновки

SQLite – реляційна СКБД, міститься в бібліотеці мови Python. Вільна, серверна та функціональна, без проблем працює з веб-додатками.

Мета дизайну SQLite полягала в тому, щоб дозволити програмі працювати без встановлення СКБД чи вимагати адміністратора БД. На відміну від багатьох інших СКБД, SQLite не є механізмом бази даних клієнт-сервер. Скоріш вона вмонтована в кінцеву програму.

SQLite є сумісною з ACID і за стандарту SQL реалізує більшу його частину. Проте вона використовує синтаксис SQL, що динамічно і слабо типізований, з цього випливає, що немає гарантії ціліснісності домену.

Головною перевагою SQLite можна вважати - нульову конфігурацію. SQLite не потрібно "встановлювати" перед його використанням. Процедура "установки" не існує. Не існує серверного процесу, який потрібно запускати, зупиняти або налаштовувати. Адміністратору не потрібно створювати новий екземпляр бази даних або призначати користувачам права доступу. SQLite не використовує файлів конфігурації. Нічого не потрібно робити, щоб повідомити системі, що SQLite працює. Все що потрібно зробити – це створити з’єднання з БД

sqlite3.connect(‘/db\_path/your.db’)