

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”**

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

КУРСОВА РОБОТА

з дисципліни “Бази даних”

спеціальність 121 – Програмна інженерія

на тему: “Екологічний моніторинг”

Студент
групи КП-01

**Аванесов Тігран
Каренович**

(підпис)

Викладач
к.т.н, доцент кафедри
СПіСКС

Радченко К.О.

(підпис)

Захищено з оцінкою _____

Київ – 2021

Анотація

Курсова робота складається з бази даних екологічних показників та консольного додатку для взаємодії з цією базою даних. Консольний додаток написано мовою програмування Python. Програма дозволяє виконувати певні операції над базою даних: зчитування, оновлення, запис, видалення, псевдовипадкова генерація записів, аналіз даних тощо.

У результаті розробки даної бази даних та даного консольного додатку було набуто практичні навички розробки сучасного програмного забезпечення, що взаємодіє з реляційними базами даних, а також здобуто навички оформлення відповідного текстового, програмного та ілюстративного матеріалу у формі проектної документації.

Зміст

Анотація	2
Зміст	3
Вступ	4
1. Аналіз інструментарію для виконання курсової роботи	5-6
2. Структура бази даних	6
3. Опис програмного забезпечення	7
3.1. Загальна структура програмного забезпечення	7
3.2. Опис основних алгоритмів роботи	7
4. Опис результатів аналізу предметної галузі	8
5. Аналіз додаткового функціоналу	8
Висновки	9
Література	10
Додатки	10-20
А. Графічні матеріали	10-13
Б. Фрагменти програмного коду	14-17

Вступ

Екологічний статус довкілля - це дуже важлива інформація, яка допомагає людям в багатьох сферах повсякденного життя. В контексті даної курсової роботи було розглянуто клімат міст України. База має такі сутності:

- Місто
- Температура повітря
- Вологість повітря
- Температура води

База даних була оптимізована шляхом індексації. Наявна функція імпорту та експорту даних у форматі txt .

1. Аналіз інструментарію для виконання курсової роботи

Для виконання даної роботи у якості системи керування базами даних було обрано PostgreSQL. Цю СУБД було обрано завдяки таким факторам:

- Відкрите ПЗ відповідає стандарту SQL - PostgreSQL - безкоштовне ПЗ з відкритим вихідним кодом.
- Підтримка великої кількості типів даних, включно з власними
- Цілісність даних з усіма необхідними обмеженнями
- Надійність, безпека
- PostgreSQL є об'єктно-реляційною СУБД, що надає певні переваги
- Сумісність з багатьма типами мереж
- Велика місткість
- Популярність даної СУБД – просто знайти вирішення потенційних проблем при розробці
- Opensource проект
- Розширення - існує можливість розширення функціоналу за рахунок своїх процедур

Для взаємодії з базою даних було обрано бібліотеку psycopg2, оскільки:

- Добре підходить для зручного використання у мові програмування Python
- Розроблена спеціально для PostgreSQL
- Найпопулярніша для взаємодії з PostgreSQL у мові програмування Python
- Має чітку, зрозумілу та вичерпну документацію з хорошими прикладами

Для візуалізації результатів аналізу даних було обрано бібліотеку tkinter, оскільки:

- Вона надає зручний інтерфейс для автоматичного будування графічних об'єктів
- Для графічних об'єктів наявна можливість дуже гнучкого налаштування з великою кількістю опцій для вигляду
- Наявна можливість будувати надзвичайно різноманітні графічні об'єкти
- Наявна чітка, зрозуміла та вичерпна документація з хорошими прикладами побудови різних графічних об'єктів з різними налаштуваннями

2. Структура бази даних

База даних має такі таблиці з полями:

1. location - локація
2. temp_air – температура повітря
3. temp_water – температура води
4. humidity_air– вологість повітря

3. Опис програмного забезпечення

3.1. Загальна структура програмного забезпечення

Розроблене програмне забезпечення містить такі компоненти:

1. База даних, що зберігає інформацію про серії, набори та мініфігурки у чотирьох таблицях
2. Засоби псевдовипадкової генерації даних
3. Засоби імпорту та експорту даних (зокрема у форматі txt)
4. Засоби аналізу даних

3.2. Опис основних алгоритмів роботи

При псевдовипадковій генерації даних для кожної таблиці процес генерації був побудований так, аби генерувалися дані у певних (нормальних) межах для обраної предметної галузі та для обраної структури бази даних.

Для генерації даних шляхом імпорту відбувається обробка вхідних даних, для забезпечення коректного їх внесення у таблиці, оскільки стиль датасетів, обраних для цієї мети, не повністю відповідає структурі бази даних.

Для аналізу безпеки довкілля було використано максимальні допустимі значення кліматичних показників (або константи)
При аналізі виводиться інформація чи є це місто безпечним / не безпечним.
Аналізована інформація експортується у форматі txt.

4. Опис результатів аналізу предметної галузі

У розробленому консольному додатку наявний такий аналіз даних, що містяться у базі:

- Аналіз екологічних показників в деяких містах України та порівняння їх з нормою

5. Аналіз додаткового функціоналу

У даній курсовій роботі при реалізації консольного додатку для взаємодії з базою даних було також реалізовано додатковий функціонал на додачу до того, який зазначений у вимогах до даного проекту, а саме можливості імпортування та експортування даних у базу даних та з неї у форматі txt.

Приклад роботи імпорту та експорту наведено у додатках.

Висновки

Під час виконання даної курсової роботи виконано таку роботу та отримано такі результати:

- Була розроблена псевдовипадкова генерація для всіх таблиць, яка генерує значення у реалістичному діапазоні
- Були розроблені засоби для аналізу даних
- Були розроблені засоби генерації записів на основі готових датасетів (імпорт) а також засоби виводу аналізованих даних (експорт) у форматі txt
- Був розроблений зручний консольний інтерфейс

У результаті виконання даної курсової роботи було набуто практичні навички розробки сучасного програмного забезпечення, що взаємодіє з реляційними базами даних, а також здобуто навички оформлення відповідного текстового, програмного та ілюстративного матеріалу у формі проектної документації.

Завдяки виконанню даної роботи було здобуто вміння розробляти програмне забезпечення для реляційних баз даних, відбулося оволодіння основами використання СУБД, а також інструментальними засобами підтримки розробки додатків для подібних баз даних.

Література

1. PostgreSQL 12.5 Documentation [Електронний ресурс] / The PostgreSQL Global Development Group // PostgreSQL: The World's Most Advanced Open Source Relational Database – Електрон. дані. – Режим доступу : <https://www.postgresql.org/docs/12/index.html>.
2. Python Graphical User Interfaces with Tk:
<https://docs.python.org/3/library/tk.html>

Додатки

А. Графічні матеріали


Структура бази даних

```
You are now connected to database "climate" as user "postgres".
climate=# CREATE TABLE climate_ind(
climate(# id_ind serial primary key,
climate(# location TEXT
climate(# temp_air FLOAT,
climate(# temp_water FLOAT,
climate(# wet_air FLOAT,
climate(# level_danger_in_air FLOAT)
climate=# CREATE TABLE
climate=#
```

Підключення через код

```
conn = psycopg2.connect(dbname = 'weather', user = 'postgres',
                        password = 'postgrepass', host = 'localhost')
cursor = conn.cursor()
conn.autocommit = True
```


Приклад аналізу

 res.txt - Notepad

File Edit Format View Help


Sevastopol is dangerous
Mykolaiv is dangerous
Zaporizhzhia is not dangerous
Melitopol is dangerous
Luhansk is dangerous
Donetsk is dangerous
Zhytomyr is dangerous
Pavlohrad is not dangerous
Odesa is dangerous
Bila Tserkva is not dangerous
Chernihiv is dangerous
Drohobych is not dangerous
Pavlohrad is dangerous
Alchevsk is dangerous
Cherkasy is not dangerous
Poltava is not dangerous
Uzhhorod is not dangerous
Khrustalnyi is not dangerous
Cherkasy is dangerous
Ternopil is not dangerous
Nikopol is dangerous
Horlivka is dangerous
Donetsk is not dangerous
Kyiv is dangerous
Lutsk is dangerous
Kremenchuk is not dangerous
Mykolaiv is not dangerous
Sloviansk is not dangerous
Kramatorsk is dangerous
Sumy is dangerous
Ternopil is dangerous
Vinnytsia is not dangerous
Simferopol is not dangerous
Rivne is dangerous
Alchevsk is not dangerous
Konotop is dangerous
Kherson is dangerous
Mariupol is not dangerous
Kamianske is dangerous
Chernivtsi is dangerous
Uzhhorod is dangerous
Lviv is not dangerous
Drohobych is dangerous
Khmelnyskyi is not dangerous

Приклад імпорту

 cities.txt - Notepad

File Edit Format View Help


Kyiv
Kharkiv
Odesa
Dnipro
Donetsk
Zaporizhzhia
Lviv
Kryvyi Rih
Mykolaiv
Mariupol
Sevastopol
Luhansk
Vinnytsia
Makiivka
Simferopol
Poltava
Chernihiv
Kherson
Cherkasy
Khmelnyskyi
Zhytomyr
Chernivtsi
Sumy
Horlivka
Rivne
Ivano-Frankivsk
Kamianske
Kremenchuk
Kropyvnytskyi
Ternopil
Luts'k
Bila Tserkva
Kramatorsk
Melitopol
Sievierodonetsk
Drohobych
Khrustalnyi
Uzhhorod
Berdiansk
Sloviansk
Nikopol
Yevpatoriia
Pavlohrad
Alchevsk
Konotop
Brovary
Kamianets-Podilskyi
Lysychansk
Dovzhansk
Mukacheve
Uman
Yalta
Yenakiieve
Shostka
Berdychiv

 const.txt - Notepad

File Edit Format View Help

temp_water:-10 to 30
temp_air:-20 to 40
humidity_air:0.1 to 8
danger_air:0.1 to 2.2

Приклад експорту

 res.txt - Notepad

File Edit Format View Help

Sevastopol is dangerous
Mykolaiv is dangerous
Zaporizhzhia is not dangerous
Melitopol is dangerous
Luhansk is dangerous
Donetsk is dangerous
Zhytomyr is dangerous
Pavlohrad is not dangerous
Odesa is dangerous
Bila Tserkva is not dangerous
Chernihiv is dangerous
Drohobych is not dangerous
Pavlohrad is dangerous
Alchevsk is dangerous
Cherkasy is not dangerous
Poltava is not dangerous
Uzhhorod is not dangerous
Khrustalnyi is not dangerous
Cherkasy is dangerous
Ternopil is not dangerous
Nikopol is dangerous
Horlivka is dangerous
Donetsk is not dangerous
Kyiv is dangerous
Lutsk is dangerous
Kremenchuk is not dangerous
Mykolaiv is not dangerous
Sloviansk is not dangerous
Kramatorsk is dangerous
Sumy is dangerous
Ternopil is dangerous
Vinnytsia is not dangerous
Simferopol is not dangerous
Rivne is dangerous
Alchevsk is not dangerous
Konotop is dangerous
Kherson is dangerous
Mariupol is not dangerous
Kamianske is dangerous
Chernivtsi is dangerous
Uzhhorod is dangerous
Lviv is not dangerous
Drohobych is dangerous
Khmelnyskyi is not dangerous

Б. Фрагменты программного коду

module.py

```
filelocals = 0
dirname = 0
fileconsts = 0
consts = 0
locals = 0
res = 0
dirname = 0
```

weather.py

```
import psycopg2
import random
import tkinter
from tkinter import *
from tkinter.ttk import *
from tkinter.filedialog import askopenfilename, askdirectory
import tkinter.font as font
import tkinter.messagebox
import module

conn = psycopg2.connect(dbname='weather', user='postgres',
                        password='postgrepass', host='localhost')
cursor = conn.cursor()
conn.autocommit = True

def danger_to_file(lst):
    fin = open(module.dirname + '/res.txt', 'w')
    for item in lst:
        fin.write(item + "\n")

def read_constants_from_file():
    fin = open(module.fileconsts, 'r')
    res = {}
    for line in fin:
```

```

        line.replace(' ', '')
        lst = line.split(':')
        lst_from_to = lst[1].split('to')
        res[lst[0]] = (float(lst_from_to[0]), float(lst_from_to[1]))
    return res

def get_lst_locals():
    fin = open(module.filelocals, 'r')
    lst = []
    for line in fin:
        lst.append(line)
    return lst

def random_input(constants):
    lst = get_lst_locals()
    for i in range(60):
        locate = (lst[int(random.uniform(0,45))]).replace("\n", "")
        temp_wat = float(random.uniform((constants['temp_water'][0]) - 1, (constants['temp_water'][1]) + 1))
        temp_air = float(random.uniform((constants['temp_air'][0]) - 1, (constants['temp_air'][1]) + 2))
        humidity_air = float(random.uniform((constants['humidity_air'][0]), (constants['humidity_air'][1]) + 1))
        danger = float(random.uniform((constants['danger_air'][0]), (constants['danger_air'][1]) + 1))
        if humidity_air < 0:
            humidity_air = 0.01
        if danger < 0:
            danger = 0.01
        cursor.execute("INSERT INTO weather_stats(location, temp_air, temp_water, humidity_air, danger_air) VALUES ('{}', '{}', {}, {}, {})".format(locate, temp_wat, temp_air, humidity_air, danger))

def analyze(constants):
    cursor.execute('SELECT * FROM weather_stats')
    id_danger = []
    for row in cursor:
        if row[1] < ((constants['temp_water'])[0]) or row[1] > ((constants['temp_water'])[1]):
            id_danger.append(str(row[5]) + " is dangerous")
            continue
        if row[2] < ((constants['temp_air'])[0]) or row[2] > ((constants['temp_air'])[1]):
            id_danger.append(str(row[5]) + " is dangerous")

```

```

        continue
    if row[3] < ((constants['humidity_air'])[0]) or row[3] > ((constants['humidity_air'])[1]):
        id_danger.append(str(row[5]) + " is dangerous")
        continue
    if row[4] < ((constants['danger_air'])[0]) or row[4] > ((constants['danger_air'])[1]):
        id_danger.append(str(row[5]) + " is dangerous")
        continue
    id_danger.append(str(row[5]) + " is not dangerous")
dangers = set(id_danger)
danger_to_file(dangers)

def clear():
    cursor.execute(f"DELETE FROM weather_stats")

def browse_const():
    module.fileconsts = askopenfilename()
    module.const['text'] = 'File containing const values was chosen'

def browse_local():
    module.filelocals = askopenfilename()
    module.locals['text'] = 'File with location names was chosen'

def browse_resp():
    module.dirname = askdirectory(parent = window, initialdir = "/", title = 'Please select a directory')
    module.res['text'] = "Directory for result output was chosen"

def start():
    clear()
    constants = read_constants_from_file()
    random_input(constants)
    analyze(constants)
    tkinter.messagebox.showinfo(title = "Success", message = "Data was analyzed and saved")

window = Tk()
window.geometry('680x550')

```



```
window.resizable(False, False)
window.title("Weather analysis")

myFont = font.Font(family = "Courier", size = 15, weight = "bold")

module.const = Label(window, text = "File containing const values wasn't chosen", font = myFont)
module.const.place(relx = 0.5, rely = 0.1, anchor = CENTER)
module.locals = Label(window, text = "File containing location names wasn't chosen", font = myFont)
module.locals.place(relx = 0.5, rely = 0.35, anchor = CENTER)
module.res = Label(window, text = "Directory for result output wasn't chosen", font = myFont)
module.res.place(relx = 0.5, rely = 0.6, anchor = CENTER)

btn_consts = Button(text = "Choose file", width = 30, command = browse_const).place(relx = 0.5, rely = 0.2, anchor =
CENTER)
btn_locals = Button(text = "Choose file", width = 30, command = browse_local).place(relx = 0.5, rely = 0.45, anchor =
CENTER)
btn_locals = Button(text = "Choose folder", width = 30, command = browse_resp).place(relx = 0.5, rely = 0.7, anchor =
CENTER)
btn_start = Button(text = "Begin analysis", width = 45, command = start).place(relx = 0.5, rely = 0.85, anchor = CENTER)

window.mainloop()
```