**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: **Система кліматичних показників довкілля**

|  |  |  |
| --- | --- | --- |
| **Студент групи** КП-93 | **Фещенко Є. О.**  (ПІБ) | \_\_\_\_\_\_\_\_\_   (підпис) |
| **Викладач**  **к.т.н, доцент кафедри СПіСКС** | **Петрашенко А.В.** | \_\_\_\_\_\_\_\_\_   (підпис) |

Захищено з оцінкою \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ – 2020

# Анотація

У процесі виконання даної курсової роботи було набуто практичні навички написання ІАС широкого призначення, з використанням аналізу/фільтрації/пагінації даних та генерації даних у реляційну БД(за допомогою 3d-party бібліотек). Результатом розробки є навички роботи з обраною СУБД та інструментами для управління даною СУБД.

Даний додаток вміє аналізувати кліматичні показники погоди(температура, вологість, дощі, сніг і тд), прогнозу погоди(очікувана температура, швидкість вітру тощо) та показники забруднення середовища (вміст шкідливих речовин в повітрі), беручи дані про міста з різних куточків світу.

Користувачу надаються можливості переглядати та генерувати дані, інші CRUD методи не були реалізовані.

Початок розробки був запланований на 30 жовтня 2020, а завершення – 13 грудня. Було присутнє невелике відхилення від плану. Далі описано подробиці виконання та проектування курсової роботи.

Галузь застосування даного програмного забезпечення - аналіз прогнозу погоди та стану довкілля.

Робота знаходиться на github репозиторії:

[study/Database Coursework at master · YehorFeshchenko/study · GitHub](https://github.com/YehorFeshchenko/study/tree/master/Database%20Coursework)

Зміст

[Анотація 2](#_Toc60111180)

[Вступ 4](#_Toc60111181)

[Аналіз інструментарію для виконання курсової роботи 5](#_Toc60111182)

[Структура бази даних 6](#_Toc60111183)

[Опис програмного забезпечення 7](#_Toc60111184)

[Загальна структура програмного забезпечення 7](#_Toc60111185)

[Опис модулів програмного забезпечення 7](#_Toc60111186)

[Опис основних алгоритмів роботи 8](#_Toc60111187)

[Аналіз функціонування засобів реплікації 9](#_Toc60111188)

[Аналіз функціонування засобів резервування/відновлення 10](#_Toc60111189)

[Аналіз результатів підвищення швидкодії запитів 11](#_Toc60111190)

[Опис результатів аналізу предметної галузі 12](#_Toc60111191)

[Висновки 13](#_Toc60111192)

[Література 14](#_Toc60111193)

[Додатки 15](#_Toc60111194)

[А. Графічні матеріали 15](#_Toc60111195)

[Б. Фрагменти коду 20](#_Toc60111196)

# Вступ

Погода сильним чином впливає на самопочуття людини. Дуже часто вона буває мінлива, і людина на своїй шкурі може відчути всі її капризи: вранці промерзнути до мозку кісток, вдень вмирати від задухи, а ввечері потрапити під справжню зливу. Саме ці чинники і змушують людей щодня перевіряти прогноз погоди, щоб знати, що їх чекає сьогодні.

Взагалі при складанні прогнозу погоди завжди головну роль відіграють такі її характеристики, як температура, вологість повітря, сила атмосферного тиску, вітер, хмарність та опади. Сьогодні найрізноманітніші технології дозволяють більш точно передбачити, яка погода чекає нас в той чи інший день. Відомості, які збираються з усіх світових метеостанцій, повітряних лайнерів, пароплавів і супутників, дозволяють побачити майже точну картину майбутнього прогнозу. Дана програма взаємодіє з відомим сервісом для відсклідковування кліматичних умов OpenWeatherMap.

На основі даних доступних лише у безкоштовній версії сервісу можна вдало аналізувати погоду.

# Аналіз інструментарію для виконання курсової роботи

У ролі СУБД виступає PostgreSQL та його графічний інтерфейс у вигляді pgAdmin 4. Дана СУБД є досить потужною і популярною для великих обсягів даних. pgAdmin 4 надає зручний користувацький інтерфейс для роботи із даною СУБД.

PostgreSQL – об’єктно-реляційна система управління базою даних. Її перевагами є її популярність, що сприяє тому, що в мережі Інтернет присутня величезна кількість інформації, в якій можна знайти всі відповіді на питання.

Також, дана СУБД має потужні механізми для реплікації та транзакції даних. Ця СУБД є не тільки реляційною, а ще й об’єктно-орієнтовною, із засобами наслідування тощо.

У ролі середовища для розробки було використано PyCharm Community Edition 2020. Дане середовище має широкий функціонал для дебагу та компіляції коду, має зручний інтуїтивно-зрозумілий графічний інтерфейс і є дуже популярною у Python ком’юніті.

Весь код був написаний мовою Python 3.8 . Це не остання версія цієї мови, але вона підтримує усі необхідні інструменти для роботи із СУБД та аналізу даних.

Також використовувалися сторонні бібліотеки та модулі, завантаженні за допомогою утиліти pip3:

* matplot та numpy для побудови графіків
* requests для здійснення веб-взаємодії із сервісом OpenWeatherMap
* psycopg2 та sqlalchemy для взаємодією з БД
* colorama для оформлення консольного інтерфейсу

# Структура бази даних

База даних складається з таких таблиць:

Таблиця City – зберігає дані про місто

Таблиця Weather – зберігає загальну характеристику погоди та дату її отримання, має зовнішній ключ, що посилається на таблицю City

Таблиця Metrics – зберігає дані кліматичних показників, має зовнішній ключ, що посилається на таблицю Weather

Таблиця Wind – зберігає показники вітру, має зовнішній ключ, що посилається на таблицю Weather

Таблиця Rain – зберігає показники дощу, має зовнішній ключ, що посилається на таблицю Weather

Таблиця Snow – зберігає показники снігу, має зовнішній ключ, що посилається на таблицю Weather

Таблиця Clouds – зберігає показники хмарності, має зовнішній ключ, що посилається на таблицю Weather

Таблиця Forecast – зберігає загальну характеристику прогнозу погоди, має зовнішній ключ, що посилається на таблицю City

Таблиця Indicators – зберігає кліматичні показники прогнозу погоди, має зовнішній ключ, що посилається на таблицю Forecast

Таблиця Air\_Pollution – зберігає дані про забрудненість повітря, має зовнішній ключ, що посилається на таблицю City

У БД присутні one-to-many та one-to-one зв’язки. Дані максимально винесені у таблиці задля збільшення швидкодії додатку.

# Опис програмного забезпечення

## Загальна структура програмного забезпечення

Програмне забезпечення побудоване по моделі MVC із рознесенням відповідних модулів.

Уся програма розділена на дві незалежних програми. Перша програма – це генератор кліматичних показників, друга – їх аналізатор. Доступ до БД реалізований за технологією ORM за допомогою бібліотеки SQLAlchemy. Також для роботи використовувалися такі бібліотеки як matprolib, psycopg2, colorama, requests.

## Опис модулів програмного забезпечення

Генератор кліматичних показників складається із файлу main.py, де здійснюється контроль над генерацією, файлами weather\_functions.py, forecast\_functions.py та air\_pollution\_functions.py куди винесена логіка обробки різних видів запитів.

Функції є універсальними, тому можливе розширення функціоналу генератору, наприклад додавання нових видів запитів. Запити реалізовані за допомогою бібліотеки requests. Дана бібліотека дозволяє виконувати інтернет запити по даному URL та отримувати сторінку у форматі HTML для подальшої роботи з ним. Необхідні URL запитів було знайдено у документаціїї до сервісу OpenWeatherMap ([Weather API - OpenWeatherMap](https://openweathermap.org/api)).

Аналізатор новин складається з 4 директорій та файлу main.py. Оскільки код написаний з використанням моделі MVC то проект відповідно має директорії model, view, controller. Директорія storage та її файл tables.py має описані за технологією ORM SQLAlchemy таблиці бази даних. Директорія model та її файл DBModel.py містить CRUD операції, описані за допомогою бібліотеки SQLAlchemy, з використанням таблиць із директорії storages.

Директорія view містить один файл view.py , який містить статичний клас із функціями для реалізації GUI через консоль. Даний модуль не включає жоден файл та бібліотеку, тому він повністю відповідає моделі MVC(у даному файлі містяться функції виводу інформації , та підтвердження дії, які викликаються контроллером як реакція на виконання роботи)

Директорія controller складається із двох файлів controller.py та controller\_functions.py. Основна логіка програми розписана у файлі controller\_functions.py, натомість у controller.py забезпечений лише необхідний виклик функцій, як реакція на дії користувача.

## Опис основних алгоритмів роботи

Як було сказано вище, у файлі controller\_functions.py прописані основні алгоритми для роботи та аналізу даних. Для виконання функціоналу CUI були створені відповідні функції, які реалізовують основну логіку запиту користувача.

Однією із основних опцій користувача є аналізи новин, тому для цього реалізовані функції аналізу даних для знаходження запропонованих варінтів запитів. Дані функції використовують інструменти numpy та matprolib для побудови необхідних даних та їх візуалізації шляхом графіків.

Реалізовано отримання даних про погоду, прогноз погоди та показники забрудненості повітря за обрану дату.

# Аналіз функціонування засобів реплікації

Реплікація була реалізована за допомгою другої інстанції власного серверу на одній Windows системі. Реплікація реалізована за принципом master-slave, для цього у PostgreSQL існує відповідна роль. Перверірка трансляції реплікації може бути виконана при зупинені сервісу postgresql-12, та запуску відповідного slave сервера.

У результаті є можливість використання slave сервера, у разі виходу з ладу основного.

# Аналіз функціонування засобів резервування/відновлення

Резервування бази даних реалізовано за допомогою стороннього програмного забезпечення SQLBackupAndFTP. Дана програма дозволяє створити з’єднання із різними СУБД, зокрема із PostgreSQL. Після підключення до необхідного серверу, вибравши необхідну для резервування БД можна вибрати опції та шлях зберігання резервування даних. Наявна можливість циклічного оновлення у обраний час. У даній роботі налаштоване резервування кожен день о 16:20, та видаленням старих резервних копій , які старіше одного місяця.

# Аналіз результатів підвищення швидкодії запитів

Оцінюючи дані та запити, було прийнято рішення використовувати B-tree індекси на поля api\_city\_id та description у таблицях City та Weather відповідно. Це пришвидчило виконання запитів (підтверджено графіками). Була спроба ввести той самий індекс на description(таблиця Air\_Pollution), але це дало зворотній ефект. Індекс B-tree формує бінарне дерево, що робить пошук по гілках за константний час. Він гарно підходить для розкиданих несортованих даних. Також він використовується у даних, які можна чітко порівняти знаками =,>,<.

# Опис результатів аналізу предметної галузі

Використовуючи розроблене програмне забезпечення було проаналізовано близько кліматичні показники та показники довкілля за останні декілька днів(з 25 грудня 2020 до 28 грудня 2020) та отримані такі результати:

Наприклад, станом на 28 грудня

----------------------------------------

У першому графіку було використано аналіз переглядів та їх швидкість, для формування найбільш популярних у даний момент часу новин.

У другому графіку (рис 7 ) використовувалася медіана переглядів по групам(темам). Даний метод аналізу має альтернативу у вигляді середнього значення. Але на відміну від останнього, медіана гарно підходить для датасету, в якому наявні критичні значення(значення які сильно відрізняються від інших).

На рисунках 8 та 9 зображена статистика авторів. При аналізі використовувалася медіана переглядів та підрахунок кількості новин.

На рисунку 10-13 було знайдено схожі новини(лідери за переглядами у своїй темі) для новини із id 738.

Дізнатись id новини можна, попередньо скориставшись списком новин, який наявний у програмі. Аналізуючи початкову новину, видно, що теми перекликаються , отже пошук відбувся коректно.

# Висновки

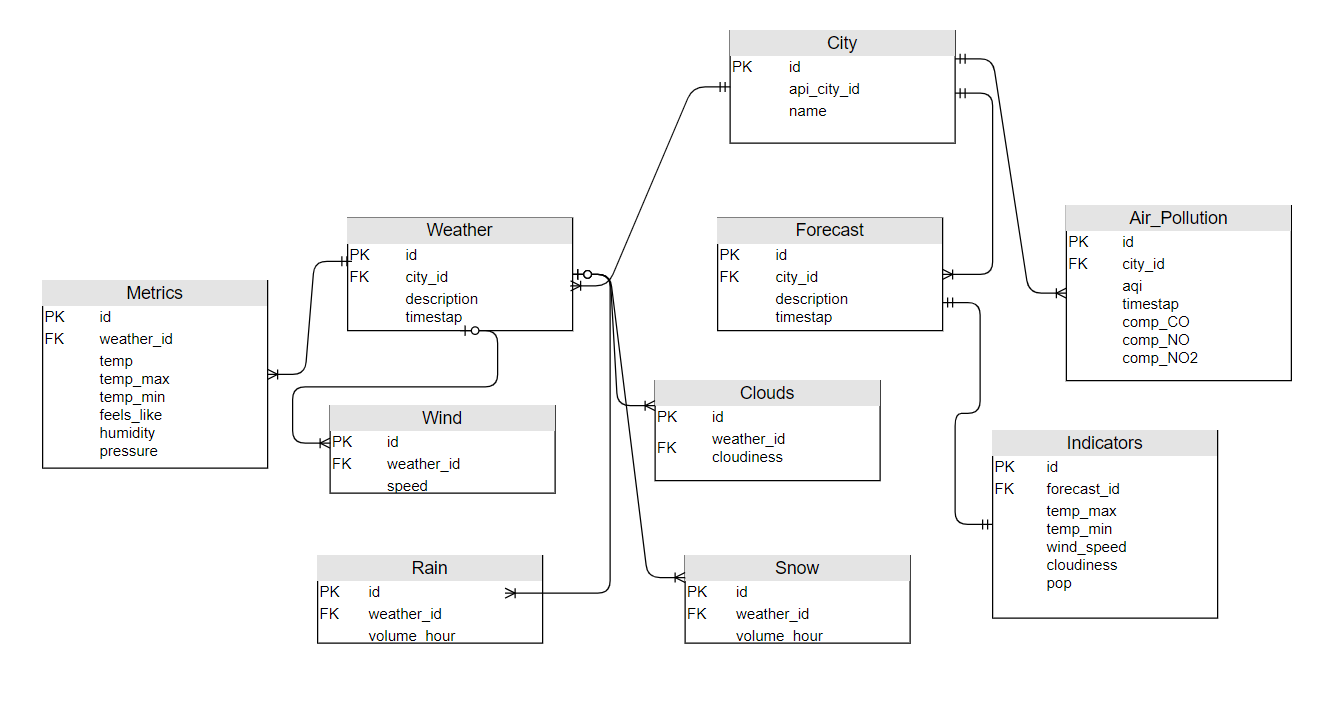
У даній курсовій роботі було розроблено програмне забезпечення для аналізу кліматичних показників довкілля, з метою їх аналізу. У процесі роботи використовувалися такі бібліотеки мови Python як matprolib, numpy, requests, sqlalchemy, psycopg2, colorama тощо. Код роботи було логічно розподілено на модулі, згідно схеми MVC(model-view-controller). Для пришвидчення виконання коду було використано інструменти індексації PostgreSQL. Для належної генерації даних у СУБД було застосовано взаємодію з API сервісу OpenWeatherMap з використанням requests. Результатом роботи даного програмного забезпечення є графіки найтепліших міст, міст з найсильнішим вітром, міст з найбільшою хмарністю, прогнози погоди з найбільшою ймовірністю осадів та міст з найбільшою концентрацією NO2.

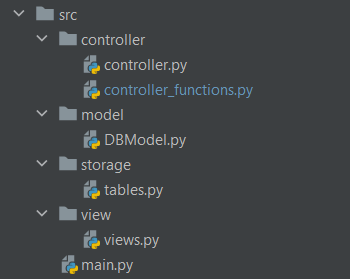
# Література

1. Data analysis using mean, median, mode [https://www.pbslearningmedia.org/resource/mwnet-math-spmmmr/data-analysis-using-mean-median-mode-and-range/](https://www.pbslearningmedia.org/resource/mwnet-math-sp-mmmr/data-analysis-using-mean-median-mode-and-range/)
2. Documentation postgreSQL <https://www.postgresql.org/docs/>
3. Python documentation <https://www.python.org/doc/>
4. Анализ данных с использованием Python <https://habr.com/ru/post/353050/>
5. Репликация баз данных PostgreSQL по типу master-slave [https://www.8host.com/blog/replikaciya-baz-dannyx-postgresql-potipu-master-slave/](https://www.8host.com/blog/replikaciya-baz-dannyx-postgresql-po-tipu-master-slave/)
6. [Weather API - OpenWeatherMap](https://openweathermap.org/api)
7. Multithreading API Requests in Python <https://creativedata.stream/multi-threading-api-requests-in-python/>
8. What is the fastest way to send 10000 HTTP requests in Python [https://stackoverflow.com/questions/43466412/what-is-the-fastestway-to-send-10-000-http-requests-in-python-2-7](https://stackoverflow.com/questions/43466412/what-is-the-fastest-way-to-send-10-000-http-requests-in-python-2-7)
9. “The clean Coder” Bob Martin
10. “Learn python 3 the hard way” Zed A. Shaw

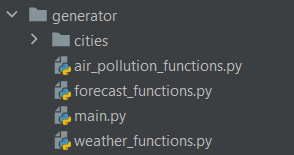
# Додатки

## А. Графічні матеріали

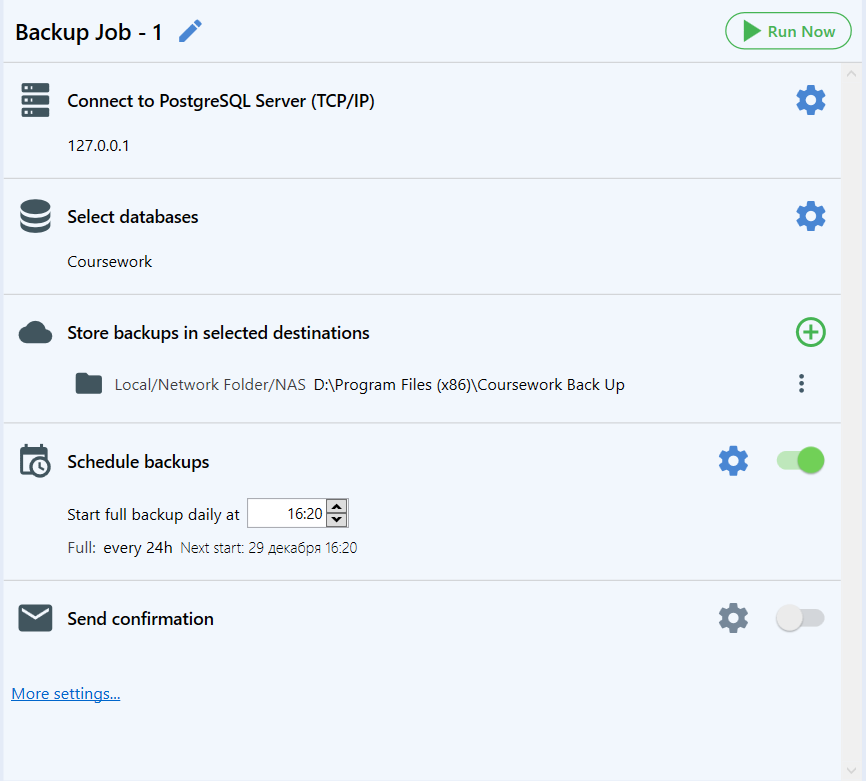


*Рис.1 Структура БД*

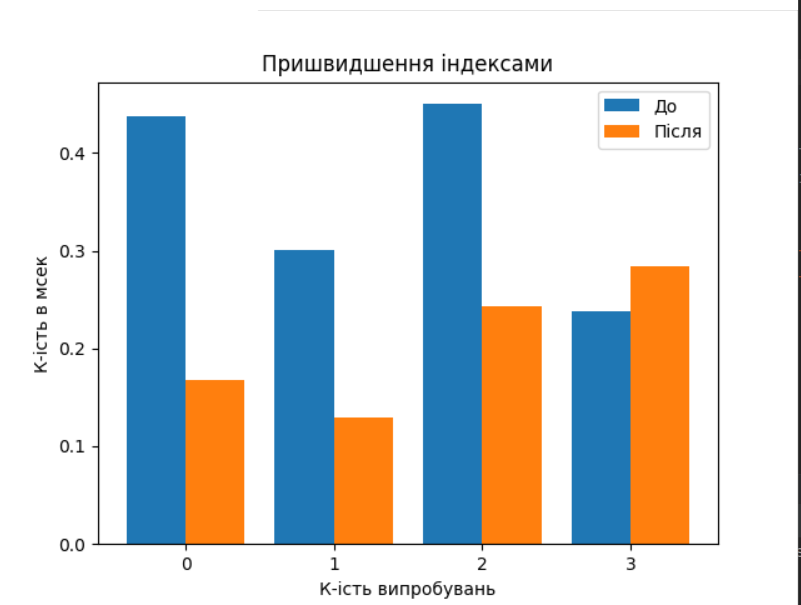
*Рис.2 Структура проекту аналізатору кліматичних показників*

******

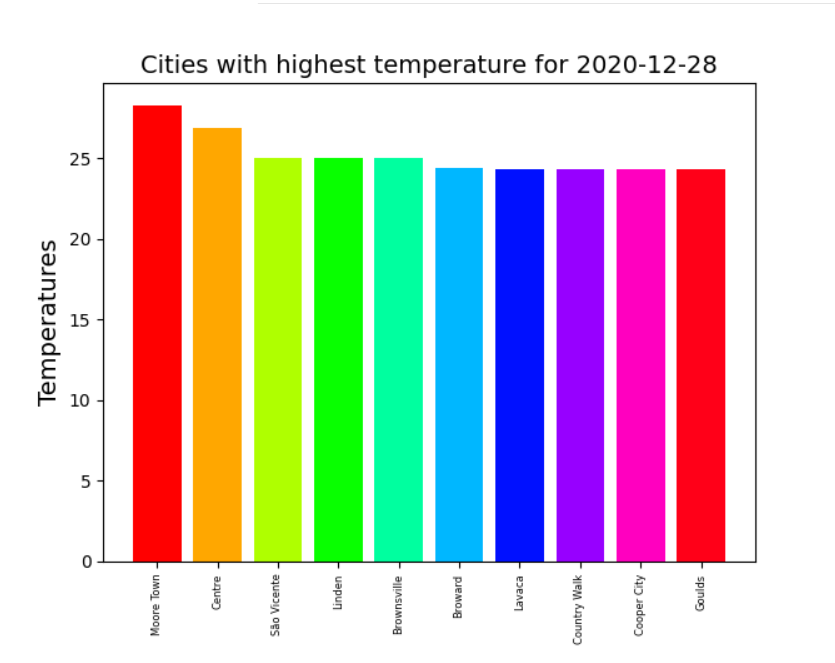
*Рис.3 Структура проекту генератора новин*

******

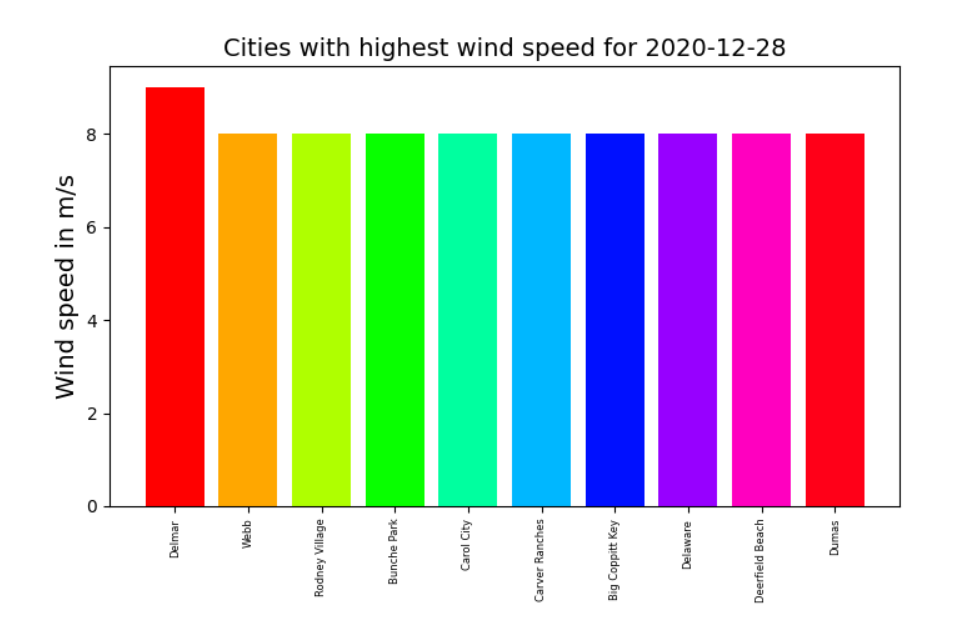
*Рис.4 Скріншот SQLBackuoAndFTP із налаштуваннями резервування*

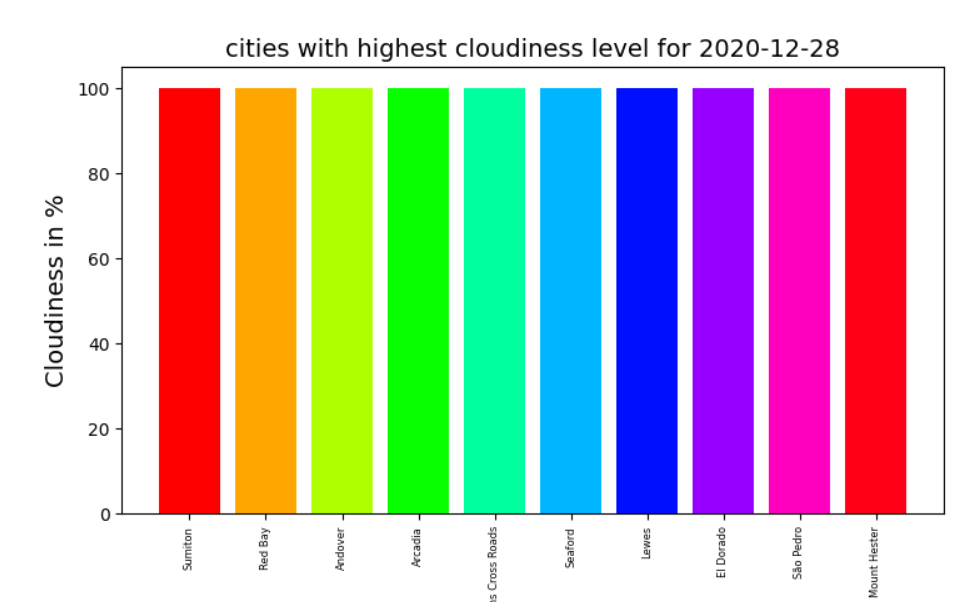
******

*Рис.5 Результати швидкодії до та після індексів*

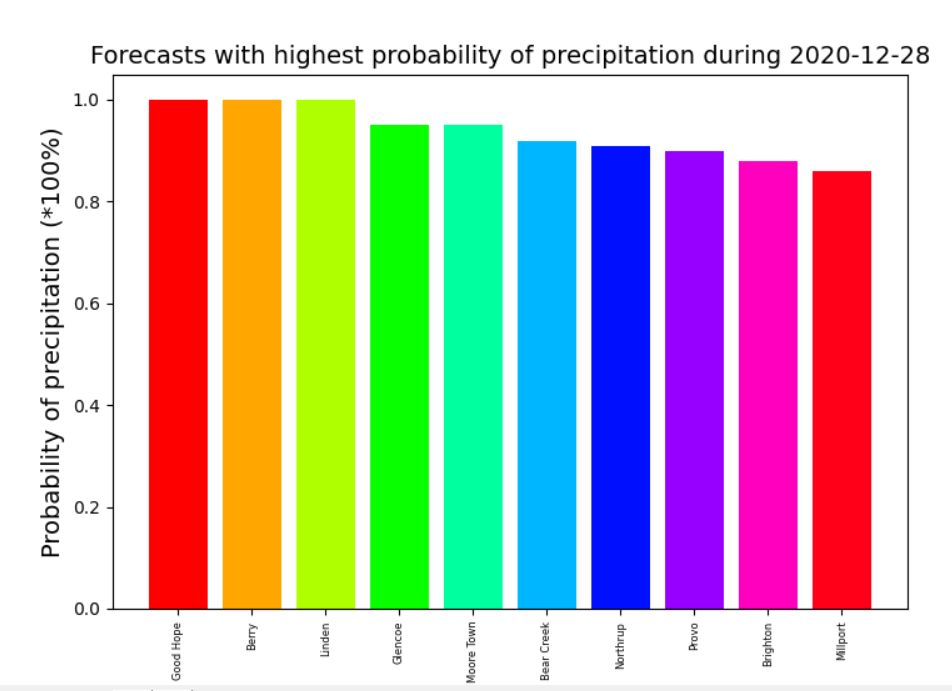
**

*Рис.6 Міста з найвищими температурами*

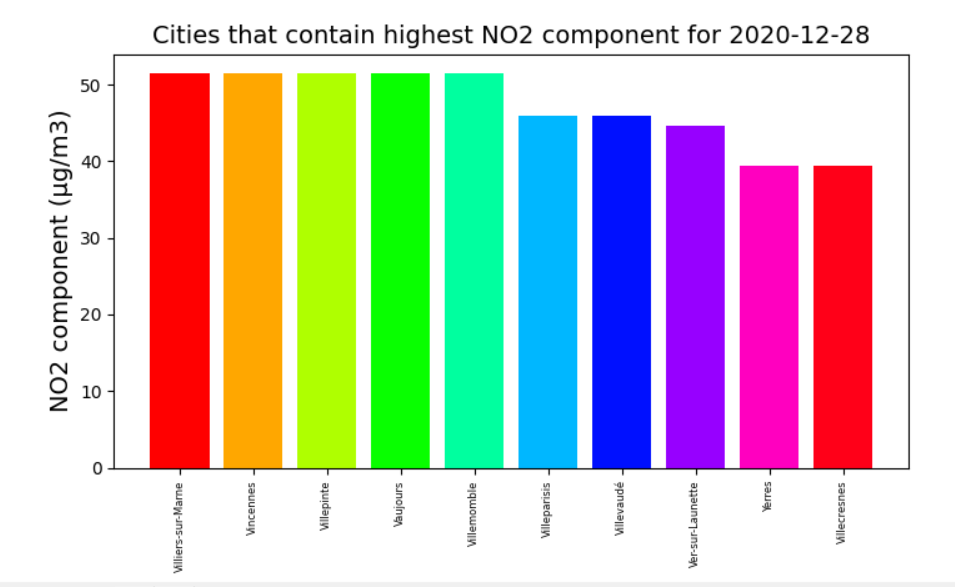
*Рис.7 Міста з найсильнішим вітром*

**

*Рис. 8 Міста з найбільшою хмарністю*

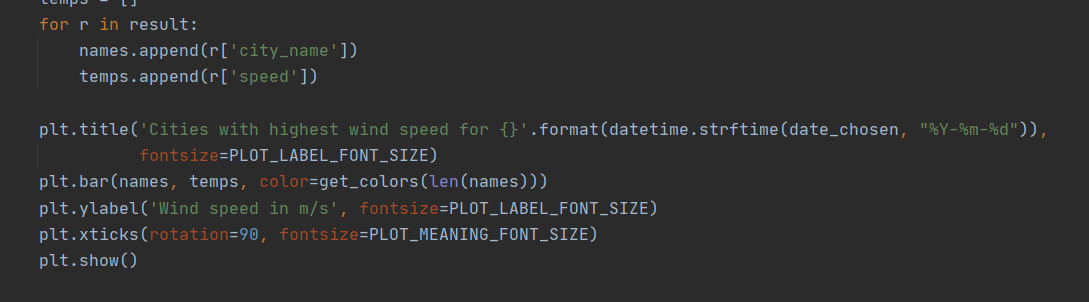
**

*Рис. 9 Прогнози погоди у містах з найвищою ймовірністю осадів*

**

*Рис. 10 Міста з найвищою концентрацією NO2*

## Б. Фрагменти коду



*Приклад використання matprolib для побудови графіків*

## 