**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: Вимірювання показників температури (води, повітря), вологості повітря, рівня шкідливих речовин у повітрі (екологічний моніторинг)

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

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

Київ – 2020

**АНОТАЦІЯ**

Дана курсова робота включала в себе здобуття практичних навичок у створенні прикладних програмних додатків, які взаємодіють із базою даних PostgreSQL. Було запропоновано наступні етапи розробки додатку:

* Створення системи отримання/генерації та фільтрації даних
* Створення системи реплікації даних
* Створення системи аналізу даних предметної галузі
* Створення системи резервування/відновлення даних

Результатом виконання курсової роботи стала реалізація усіх пунктів, описаних вище, та отримання кінцевої інформаційно-аналітичної системи, яка виконує аналіз даних, отриманих із зовнішніх ресурсів, та взаємодіє із реляційною СУБД PostgreSQL.

**ЗМІСТ**

Анотація……………………………………………………………………………..2

Зміст……………………………………………………………………………….…3

Вступ……………………………………………………………………………........4

1. Аналіз інструментарію для виконання курсової роботи………………...…....5
2. Структура бази даних……………………………………………………...……6
3. Опис програмного забезпечення………………………………………...……..7
   1. Загальна структура програмного забезпечення…………………..……7
   2. Опис модулів програмного забезпечення………………………..…….7
   3. Опис основних алгоритмів роботи……………………………….…….7
4. Аналіз функціонування засобів реплікації……………………………...……..8
5. Аналіз функціонування засобів резервування/відновлення…………...…….10
6. Аналіз результатів підвищення швидкодії запитів……………………...…..13
7. Опис результатів аналізу предметної галузі……………………………...… 14

Висновки………………………………………………………………………...…15

Література………………………………………………………………………….17

Додатки………………………………………………………………………….….18

**ВСТУП**

Основною метою даної курсової роботи є створення інформаційно-аналітичної системи за допомогою мови програмування Python у середовищі розробки PyCharm, яка взаємодіє із базою даних PostgreSQL.

Призначенням програмного додатку є здійснення аналізу показників температури (води, повітря), вологості повітря, рівня шкідливих речовин у повітрі (екологічний моніторинг).

Створення інформаційно-аналітичного додатку є актуальною задачею, оскільки він дозволяє значно прискорити, а також забезпечити процес аналізу великих обсягів даних, неможливий для здійснення людиною.

У процесі розробки було постановлено наступні основні етапи:

* Створення системи генерації/отримання та фільтрації даних, основним призначенням якої є генерація/отримання даних для подальшої їх фільтрації та аналізу.
* Створення системи реплікації даних, яка передбачає розділення функціональності отримання даних та їх передачі програмному додатку, для розподілення навантаження на базу даних та зменшення можливості втрати даних.
* Створення системи аналізу даних, яка необхідна для аналізу отриманих раніше даних та їх візуалізації у вигляді графіків та діаграм.
* Створення системи резервування/відновлення даних, необхідної для збереження стану бази даних та можливості відновлення даних на основі обраного користувачем файлу стану.

Усі ці пункти є важливими для програмного додатку, оскільки забезпечують стабільну роботу системи, а також надійність збереження даних.

Окрім пунктів, наведених вище, варто відзначити і процес підвищення швидкодії отримання даних із бази даних на основі застосування індексів БД, який також було виконано.

**І. АНАЛІЗ ІНСТРУМЕНТАРІЮ ДЛЯ ВИКОНАННЯ КУРСОВОЇ РОБОТИ**

Для виконання курсової роботи було обрано наступний інструментарій:

* СУБД – PostgreSQL 11. Даний вибір базується на тому, що PostgreSQL має безліч можливостей: тригери, індекси, функції, вбудовані методи реплікації та резервування. Крім того, дана СУБД дозволяє розгортання БД на декількох машинах та має широку підтримку.
* Мова програмування – Python 3.9. Ключовим фактором у виборі мови програмування був її широкий спектр застосування у сфері Data Science та Big Data завдяки широкому спектру бібліотек.
* Середовище розробки – PyCharm, оскільки дане середовище забезпечує комфортний процес написання коду, має вбудовані засоби його налагодження та виконання. Крім того, PyCharm має комфортний інтерфейс для пошуку та встановлення необхідних бібліотек.
* Засоби генерації даних – отримання даних виконувалося шляхом здійснення requests запитів до Visual Crossing Weather API. Даний метод генерації даних забезпечує отримання достовірних даних і отримання адекватної статистики у подальшому при аналізі.
* Засоби для роботи з СУБД – psycopg2 та sqlalchemy. Перша бібліотека є написаною на мові С, в результаті чого є ефективною та надійною[1] при завантаженні великих обсягів даних (отримання даних із API), тим часом як sqlalchemy  надає повний набір добре відомих шаблонів корпоративного рівня стабільності, сконструйованих для високопродуктивного доступу до бази даних[2].
* Засоби аналізу даних – Pandas. Ця бібліотека написана на основі бібліотеки numpy, реалізованої мовою програмування С[3], що забезечує високу її високу швидкодію при аналізі великих обсягів даних.
* Візуалізація даних – бібліотека matplotlib.

**ІІ. СТРУКТУРА БАЗИ ДАНИХ**

База даних містить у собі наступні таблиці:

* Details – для опису всіх показників погоди і їх умови, містить наступні поля id\_det, min\_tem, max\_tem, wind\_speed, wind\_dir, dew\_point, precipitation, visibility, conditions, could\_cov, relative\_hum.
* Sity Date – сутність для опису країн і їх міст в певний час і деталями погоди, містить наступні поля id\_cd, id\_loc, id\_det, date.
* Location ­– для опису сутності локації, містить наступні поля id\_loc, address, longitude, latitude.

Зв`язки між сутностями:

* Location – Sity Date: ONE to MANY
* Details – Sity Date: ONE to MANY

**ІІІ. ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

1. **Загальна структура програмного забезпечення**

Програмний додаток було реалізовано у консольному варіанті із застосуванням шаблону ORM на базі операційної системи Windows 10.

1. **Опис модулів програмного забезпечення**

Програмний додаток складається із наступних модулів:

1. Models – директорія, яка містить моделі для роботи з ORM:

* City Date
* Location
* Details

1. View – модуль для генерування виду консольного меню для взаємодії з користувачем.
2. Controller – виконує збір даних шляхом запитів до модуля database і view.
3. Api\_data – модуль, який виконує запити до Visual Crossing Weather API з метою отримання актуальних даних, та передає в database для запису.
4. Database – модуль, який виконує функції з контролера для отримання даних з slave-сервера або запису у master-сервер.
5. Main – ядро програмного додатку.
6. Data – модуль забезпечує фільтрацію та аналіз зібраних даних.
7. **Опис основних алгоритмів роботи**

Данні беруться за допомогою Visual Crossing Weather API, так як це найкращий сервер, де є все, що потрібно для отримання детальних даних у будь-якій точці світу і з необхідною точністю.

Для аналізу двох країн був використаний алгоритм, який бере три дані за день і визначає середнє арифметичне значення і зрівнює цей же день з іншою країною підставляючи це у графік для детального спостереження.

Інший аналіз бере всі країни за даним проміжком і вираховує який процент демонструючи на діаграмі результати.

Робота програмного додатку базується на взаємодії модулів та алгоритмах, реалізованих у бібліотеці pandas. У разі виникнення помилок використані можливості перехоплення помилок за допомогою конструкції try…except.

**IV. АНАЛІЗ ФУНКЦІОНУВАННЯ ЗАСОБІВ РЕПЛІКАЦІЇ**

PostgreSQL дозволяє реалізацію реплікації за допомогою наступних методів:

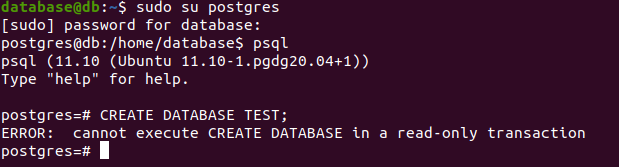
* Трансляція файлів (головний сервер працює в режимі постійної архівації змін, тим часом як кожен резервний сервер виконує постійне отримання даних у вигляді WAL файлів від головного сервера).
* Потокова реплікація (працює таким ж самим чином, як і при трансляції файлів, проте резервний сервер може працювати із меншими затримками, ніж у першому випадку. Резервний сервер підключається до головного, який передає для нього потік записів WAL у момент їх додавання, не чекаючи на їх повне заповнення)
* Каскадна реплікація (резервний сервер приймає підключення реплікації та потоки WAL від інших резервних серверів, які виступають посередниками).
* Синхронна реплікація (при такому методі реплікації кожна фіксація транзакції очікує на підтвердження того, що транзакція була записана у журнал транзакції на обох серверах: головному та резервному) [4].

У процесі виконання курсової роботи було обрано другий варіант реалізації

реплікації, реалізований за допомогою вбудованих можливостей СУБД.

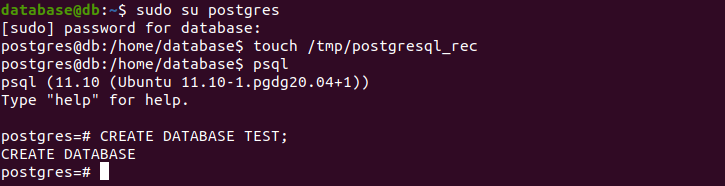
Перевіримо роботу реплікації при виході з ладу головного сервера:

1. Перевіримо наш резервний сервер на коректність роботи:



Як бачимо, можливість створення нової бази даних на ньому заблокована, отже, робота цього сервера коректна.

1. Створимо трігер файл за шляхом /tmp/postgresql\_rec, як було вказано у файлі recovery.conf при налаштуванні реплікації, та перевіримо роботу резервного сервера вдруге:



Резервний сервер почав виконувати роль головного сервера.

1. Після того, як роботу старого головного сервера було налагоджено, виконуємо його конфігурування, як резервного сервера, надаючи йому підключення реплікації до нового головного сервера.

**V. АНАЛІЗ ФУНКЦІОНУВАННЯ ЗАСОБІВ РЕЗЕРВУВАННЯ/ВІДНОВЛЕННЯ ДАНИХ**

СУБД PostgreSQL пропонує наступні методи реалізації резервування та відновлення даних:

* Використання періодичного резервного копіювання за допомогою вбудованої утиліти pg\_dump.
* Резервне копіювання на основі базових копій та архівів WAL.

Перший метод є достатньо швидким і простим у реалізації, проте забезпечує

виконання відновлення лише за наявності backup-файла, тим часом як другий метод є складнішим у реалізації, потребує більше ресурсів та пам’яті [5].

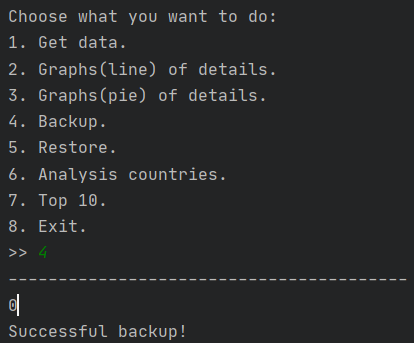
У процесі розробки програмного додатку було застосовано перший метод резервування та відновлення, що забезпечує вибір користувачем файлу резервної копії та відновлення у будь-який із раніше збережених станів.

У програмному додатку резервне копіювання та відновлення можуть бути використані у разі виникнення наступних ситуацій:

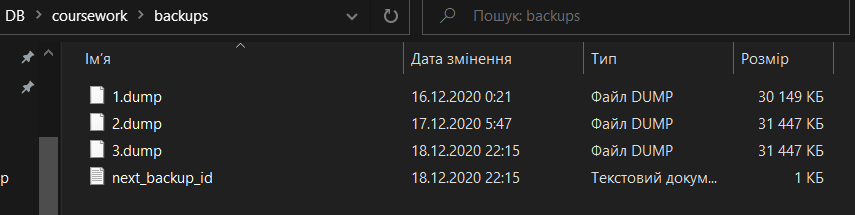
* У випадку втрати як даних бази, так і самої бази
* При виникненні потреби відновити один з попередніх станів сховища даних, збережених раніше

Наведемо приклади використання описаної функціональності:

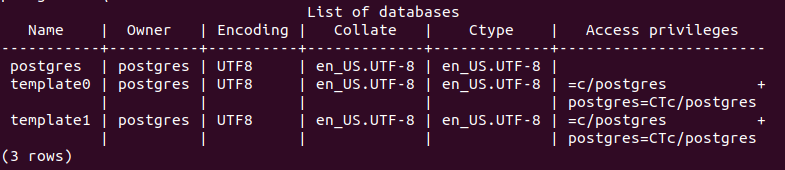
1. Створимо файл відновлення поточного стану бази даних:



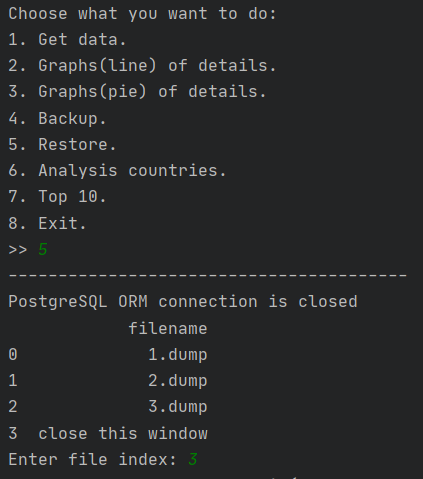
При перевірці директорії із файлами відновлення бачимо, що відповідний файл було створено:

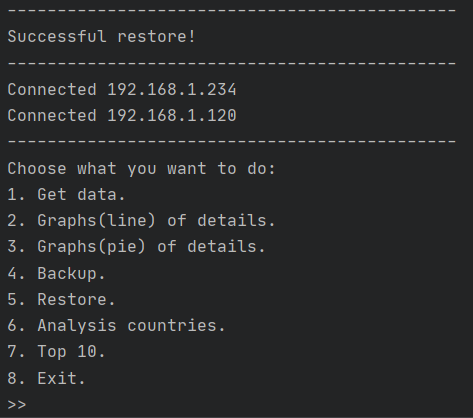


1. Тепер виконаємо відновлення. Припустимо, що базу даних було втрачено:

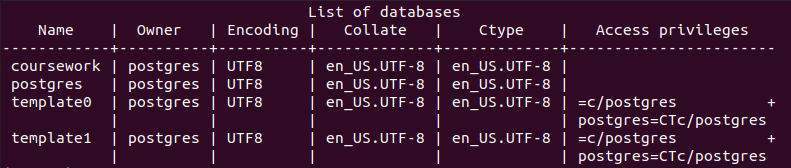


Запустимо відновлення за допомогою програми:





Втрачені дані було відновлено:



**VI. АНАЛІЗ РЕЗУЛЬТАТІВ ПІДВИЩЕННЯ ШВИДКОДІЇ ВИКОНАННЯ ЗАПИТІВ**

SELECT запити мають нижчу швидкодію при збільшенні обсягів даних та фільтрації за полями, які не входять до складу PK. Це спричинено тим, що у такому випадку виконується послідовне сканування усіх даних, що є досить неефективним. Саме тому у таких випадках (в середньому при наявності більш ніж 10000 рядків сутностей) застосовується індексація.

У таблиці Details було використано btree індекс для поля conditions, внаслідок чого вдалося отримати приріст швидкодії при отриманні умов погоди. У середньому, час планування операції зазнав або незначного приросту, або зменшення, тим часом як час виконання операції став у середньому меншим за 1 секунду. Візуальне представлення цих даних було наведено у додатку.

**VII. ОПИС РЕЗУЛЬТАТІВ АНАЛІЗУ ПРЕДМЕТНОЇ ГАЛУЗІ**

Аналіз даних предметної галузі було виконано для наступних сценаріїв:

* Аналіз будь-якого поля з сутності Details представлений у вигляді графіка.
* Аналіз будь-якого поля з сутності Details представлений у вигляді діаграми.
* Вивід у вигляді стовбчастих діаграм порівняння двох країн на будь-яке поле з сутності Details, для деталізації абсолютної статистики.
* Вивід діаграми статистики TOP 10 країн за останній місяць будь-якого поля з сутності Details.

Візуальне представлення отриманих результатів аналізу наведено у додатку.

**ВИСНОВКИ**

Підіб`ємо підсумки по кожному з етапів виконання курсовою роботи:

* Для виконання курсової роботи було підібрано вдалий інструментарій у вигляді різних програмних бібліотек та програмного забезпечення, що дало змогу реалізувати усі програмні пункти роботи.
* Структуру бази даних було спроектовано таким чином, щоб вона відповідала необхідним нормальним формам та забезпечувала найоптимальніший шлях для отримання даних із зовнішнього API.
* Програмний додаток було реалізовано за допомогою мови програмування Python 3.9 на базі операційної системи Windows 10. Програмний код відповідає шаблону проектування ORM, завдяки чому його було розбито на невеликі та компактні модулі. Надлишкова або невідповідна функціональність в модулях відсутня.
* Було проаналізовано усі засоби реплікації, доступні у СУБД PostgreSQL, внаслідок чого механізм реплікації було реалізовано за допомогою потокового методу, що забезпечує швидку асинхронну взаємодію між головним та резервним серверами. Тестування реплікації показало, що вона працює належним чином та забезпечує швидкий процес налагодження системи у разі відмови головного сервера.
* Внаслідок аналізу методів резервування/відновлення було обрано метод, який забезпечує менше використання ресурсів системи та дозволяє вибір точки відновлення серед створених користувачем раніше. У відповідному пункті було проаналізовано роботу відповідного механізму та підтверджено його коректну роботу.
* За допомогою механізму індексації в PostgreSQL вдалося отримати підвищення швидкодії виконання запитів.
* Було реалізовано механізм аналізу та візуалізації даних у вигляді діаграм та графіків за допомогою зовнішніх бібліотек мови програмування Python, таких як pandas, matplotlib, що забезпечує швидку роботу даними та їх аналіз.
* Засоби генерації та фільтрації даних було реалізовано за допомогою звернень до зовнішнього серверу даних Visual Crossing Weather API, що дозволяє отримання достовірних для обраної предметної галузі даних та їх оновлення.

У процесі виконання курсової роботи було набуто практичних навичок у створенні програмних додатків мовою програмування Python, які взаємодіють із реляційною СУБД, здобуто практичний дослід технологіях аналізу даних.

Відповідно до наведених виконаних пунктів роботи можна стверджувати, що основна мета роботи (вимірювання показників температури (води, повітря), вологості повітря, рівня шкідливих речовин у повітрі) була досягнута – додаток готовий до використання.

**ЛІТЕРАТУРА**

1. What is PostgreSQL? educba: веб-сайт. URL: <https://www.educba.com/what-is-postgresql/>
2. Advantages of PostgreSQL. bitnine: веб-сайт. URL: <https://bitnine.net/blog-postgresql/advantages-of-postgresql/?ckattempt=1>
3. Advantages of PostgreSQL. Cybertec PostgreSQL: веб-сайт. URL: <https://www.cybertec-postgresql.com/en/postgresql-overview/advantages-of-postgresql/>
4. PyGreSQL vs psycopg2. stackoverflow: веб-сайт. URL: <https://stackoverflow.com/questions/413228/pygresql-vs-psycopg2>
5. What is the best python module to use with PostgreSQL?<https://www.quora.com/What-is-the-best-python-module-to-use-with-PostgreSQL>
6. TOP 5 PYTHON LIBRARIES FOR VISUALIZATION. Tech Shenanigans: веб-сайт. URL:

<https://www.techshenanigans.com/post/top-5-python-libraries-for-visualization>

1. How Not To Sort By Average Rating. Evanmiller: веб-сайт. URL:

<https://www.evanmiller.org/how-not-to-sort-by-average-rating.html>

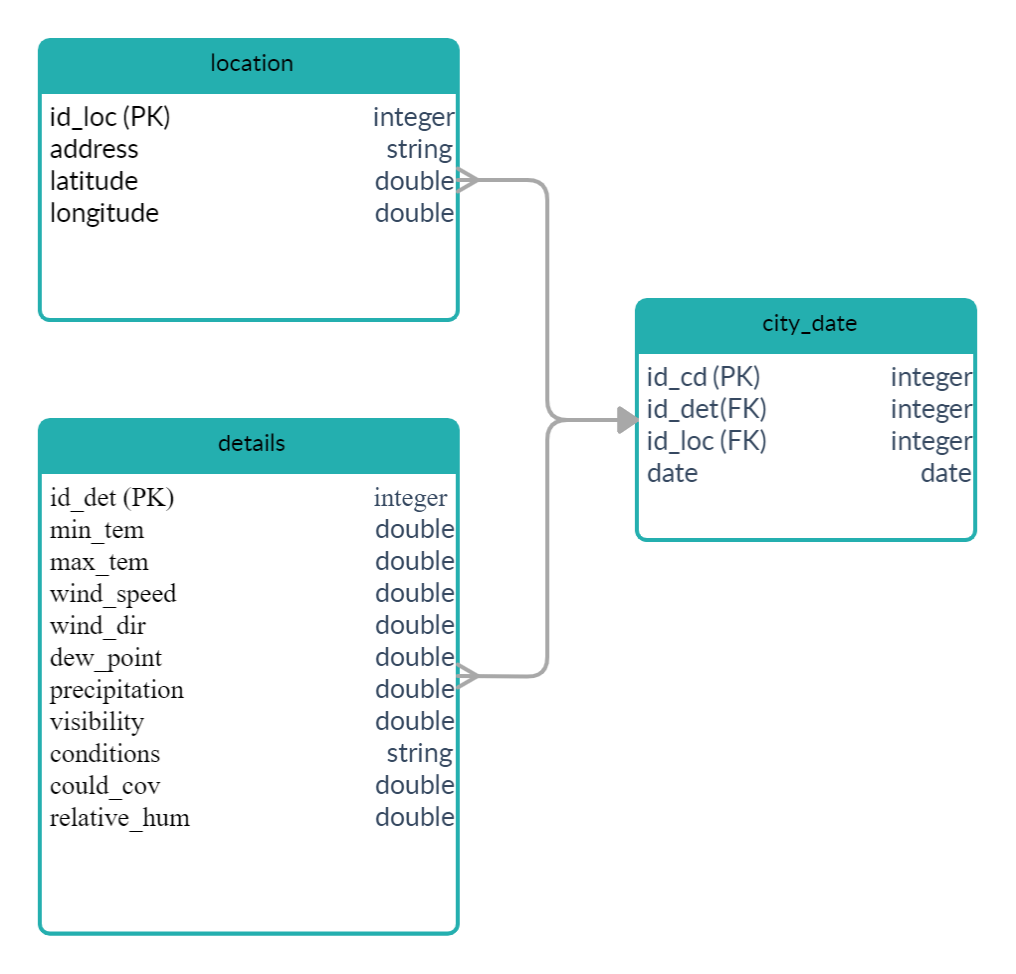
1. Bayesian Average Ratings. Evanmiller: веб-сайт. URL:

<https://www.evanmiller.org/bayesian-average-ratings.html>

1. Резервное копирование и восстановление в PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/178567/>.
2. Введение в pandas: анализ данных на Python [Електронний ресурс] – Режим доступу до ресурсу: <https://khashtamov.com/ru/pandas-introduction/>.
3. PostgreSQL: Документация [Електронний ресурс] – Режим доступу до ресурсу: https://postgrespro.ru/docs/postgresql/9.6/warm-standby.

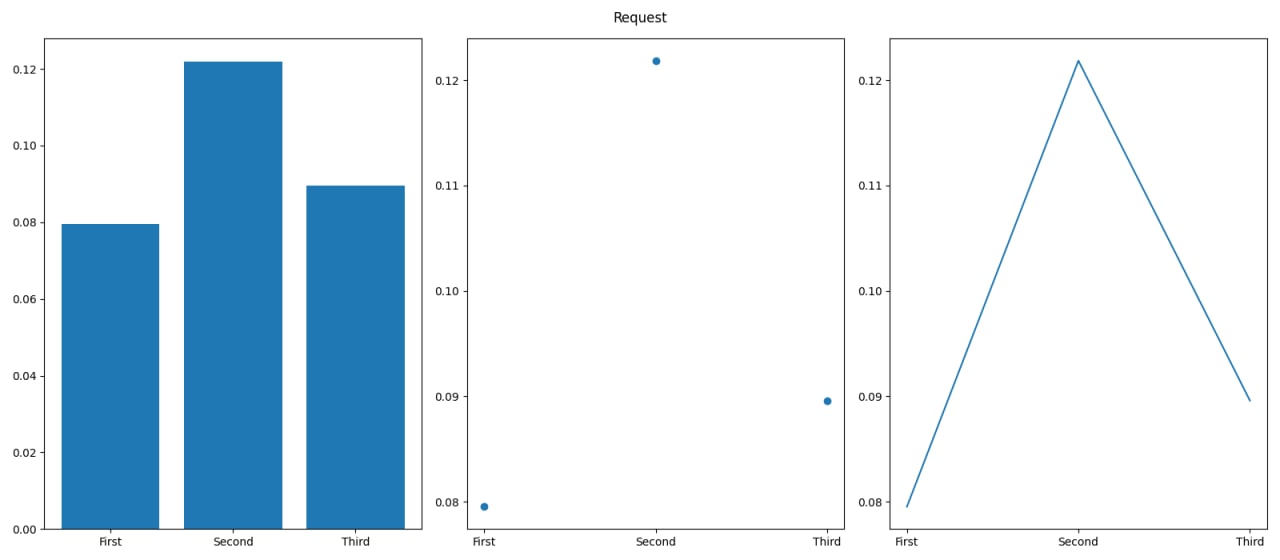
**ДОДАТКИ**

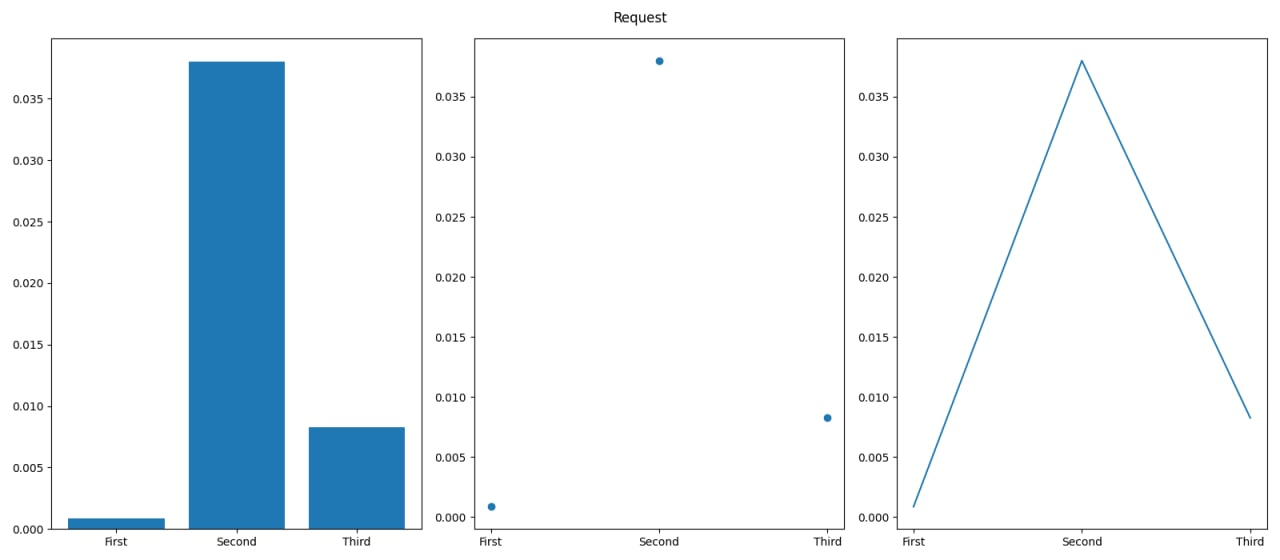
Структура бази даних:



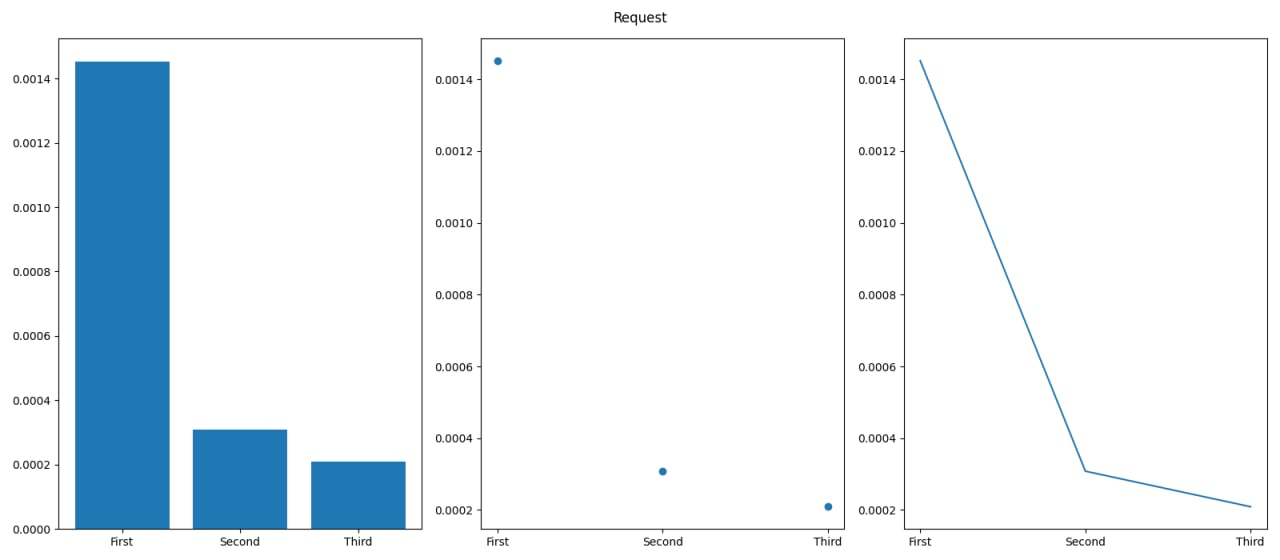
Результати підвищення швидкодії:

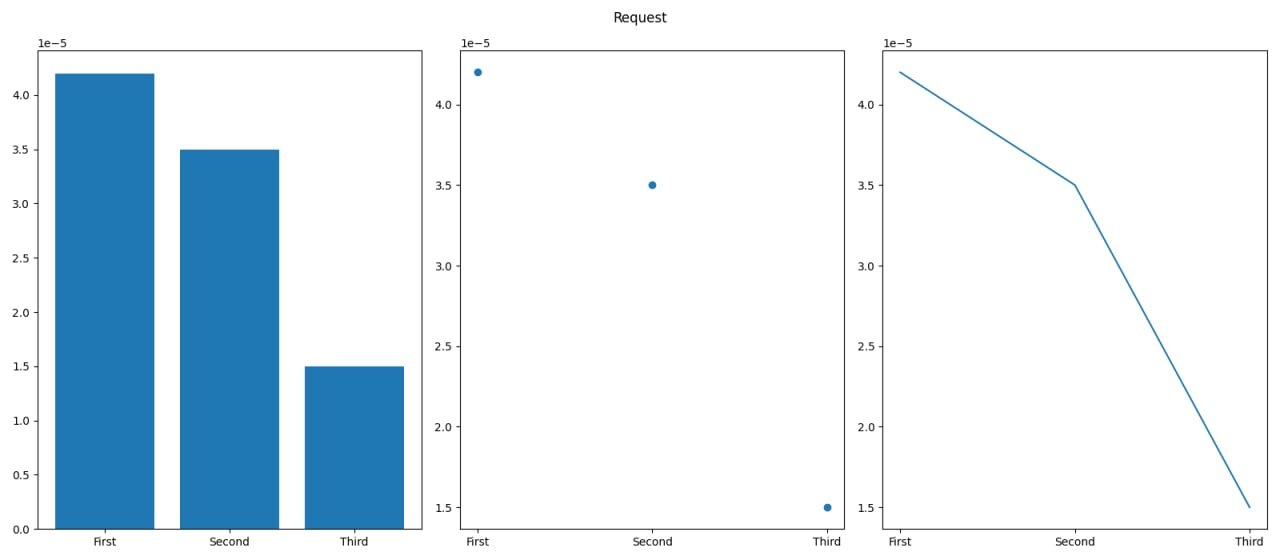
До:





Після:





Результати аналізу даних:

