Київський національний університет імені
Тараса Шевченка
Факультету комп'ютерних наук та кібернетики

Лабараторна робота №2

З дисципліни «Сучасні технології баз даних»

Виконав студент 3-го курсу
Групи ТТП-32 Пелих Олександр

Київ-2024

Тема лабараторної роботи

Дослідження ефективності використання NoSQL баз даних у порівнянні з реляційними базами (SQL)

Опис проекту

Цей проект спрямований на дослідження ефективності використання NoSQL баз даних порівняно з традиційними реляційними базами даних (SQL) у контексті зберігання та обробки даних. Метою є вивчити, чи можуть NoSQL бази, зокрема MongoDB, бути ефективнішою альтернативою для деяких типів завдань, порівняно з реляційними базами даних, такими як PostgreSQL.

Проект надає практичний інструмент для порівняння швидкодії та зручності використання NoSQL і реляційних баз даних, а також показує, як їх можна застосовувати для реальних задач з огляду на різні вимоги щодо зберігання і обробки даних.

Реалізація

1) Структура роботи

/LAB-2

│

├── main.py # *Основний файл для виконання тестів та вимірювання швидкодії*

├── mongo_db/

│ └── mongo_controller.py # *Контролер для MongoDB, який містить функції для CRUD операцій та резервного копіювання*

└── postgres_db/

└── postgres_controller.py # *Контролер для PostgreSQL, що реалізує CRUD операції та резервне копіювання*.

2) Опис файлів роботи:

- main.py — головний скрипт, який відповідає за взаємодію з базами даних, викликаючи методи з контролерів MongoDB і PostgreSQL. У ньому вимірюється час виконання операцій та проводиться тестування швидкодії (CRUD операції, експорт та резервне копіювання даних).

- mongo_db/mongo_controller.py — клас, який містить методи для роботи з MongoDB, зокрема для виконання CRUD операцій, експорту даних у JSON та резервного копіювання.

- postgres_db/postgres_controller.py — клас для роботи з PostgreSQL, що реалізує аналогічні функції для запису, читання, оновлення та видалення даних, а також функції для експорту даних у CSV та резервного копіювання.

3) Функціональні можливості

У проекті було реалізовано наступні функціональні можливості для MongoDB та PostgreSQL:

1. CRUD операції (Create, Read, Update, Delete):
   - Для кожної з баз даних створено методи, які дозволяють додавати нові записи (створення), оновлювати існуючі (оновлення), отримувати дані (читання) та видаляти записи (видалення).
   - MongoDB використовує колекції для зберігання документів, тоді як PostgreSQL працює з таблицями для організації даних.

2. Експорт даних:
   - Для MongoDB реалізовано функцію експорту колекцій у формат JSON, що дозволяє зберігати дані у зручному для подальшої обробки вигляді.
   - Для PostgreSQL дані експортуються у CSV формат, що є стандартом для реляційних баз даних і дозволяє зручно працювати з даними поза системою.

3. Резервне копіювання:
   - Для MongoDB реалізовано функцію резервного копіювання, яка створює бекап у форматі JSON, зберігаючи дані з колекції.
   - Для PostgreSQL резервне копіювання здійснюється за допомогою утиліти pg_dump, яка створює дамп бази даних у вигляді файлу.

4. Тестування швидкодії:
   - Для кожної операції (створення, оновлення, видалення, експорт та резервне копіювання) вимірюється час виконання, що дозволяє порівняти їх ефективність.
   - Тести проводяться для різної кількості записів, щоб визначити, як швидко обробляються дані в обох типах баз на різних етапах.

4) Тестування

4.1) Опис тестових операцій

- CRUD Операції:

  Для кожної бази даних було виконано операції створення, читання, оновлення та видалення (CRUD) для 100 записів. Це дозволило виміряти ефективність обробки основних операцій з даними в обох системах.

- Експорт даних:

  Дані з обох баз даних було експортовано у формати JSON (для MongoDB) і CSV (для PostgreSQL) для вимірювання часу, необхідного для експорту даних.

- Резервне копіювання:

  Для кожної бази даних було виконано операцію резервного копіювання, щоб оцінити час, необхідний для створення бекапів.

4.2) Результати тестування

```
Mongo CRUD 100 records took 2.34 seconds.
Postgres CRUD 100 records took 4.12 seconds.

Exporting from Mongo took 1.23 seconds.
Exporting from Postgres took 1.56 seconds.

MongoDB backup took 0.89 seconds.
PostgreSQL backup took 1.23 seconds.
```

4.3) Висновки тестування

- Продуктивність MongoDB у виконанні CRUD операцій і експорту даних виявилась кращою порівняно з PostgreSQL. Ці результати можуть свідчити про те, що MongoDB є більш ефективним для виконання великих обсягів операцій з документами.
- Час резервного копіювання в MongoDB також був меншим, що може бути важливим фактором для вибору MongoDB у сценаріях, де потрібна швидка та ефективна створення бекапів.
- Загальна продуктивність MongoDB показала кращі результати на всіх етапах тестування, однак варто зазначити, що обидві бази даних мають свої переваги та можуть бути вибрані залежно від вимог проекту.

Висновок

У ході виконання лабораторної роботи було проведено порівняння продуктивності двох популярних систем керування базами даних — MongoDB та PostgreSQL. Для цього було здійснено кілька основних тестів, які дозволили оцінити швидкість виконання різних операцій з даними. Зокрема, було виміряно час, витрачений на:

- CRUD операції — вставка, оновлення, видалення та читання 100 записів у кожній з баз даних.
- Експорт даних з MongoDB у формат JSON і з PostgreSQL у формат CSV.
- Резервне копіювання даних у кожній з баз.

MongoDB демонструє кращу продуктивність у всіх аспектах тестування порівняно з PostgreSQL, зокрема у виконанні CRUD операцій та експорті даних. Однак варто зазначити, що результати можуть залежати від конкретних налаштувань баз даних, розміру даних та використовуваних методів. PostgreSQL, хоч і трохи поступається MongoDB в даному тесті, залишається дуже потужною системою для реляційних даних і має свої переваги у разі складних запитів та транзакцій.

Таким чином, вибір між MongoDB та PostgreSQL залежить від конкретних вимог до проекту. MongoDB підходить для сценаріїв з великими обсягами нереляційних даних та високою швидкістю операцій, тоді як PostgreSQL краще підходить для складних реляційних запитів і транзакцій, що потребують консистентності та високої надійності.