# Лабораторна робота №7

**Тема:** Поглиблене вивчення MongoDB: оптимізація продуктивності, використання шардінгу та реплікації, інтеграція з Pandas та Machine Learning

**З дисципліни:** Бази даних та інформаційні системи

**Студента групи МІТ-31:** Циби Ярослава

# Хід роботи

**Частина 1: Оптимізація продуктивності запитів**

Для початку роботи було встановлено розширення MongoDB for VS Code.

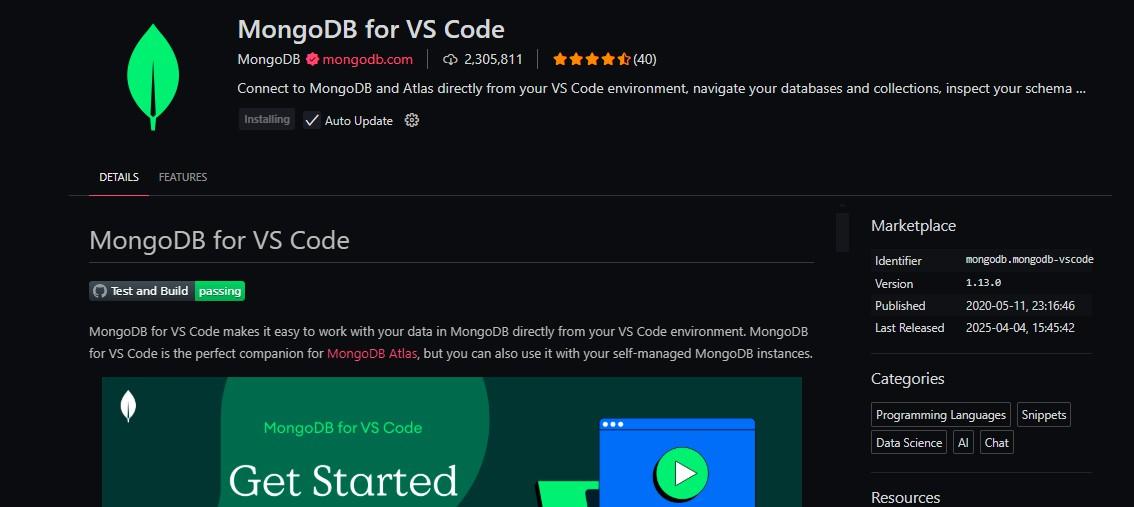


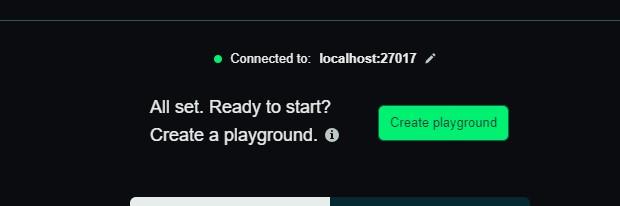
Рисунок 7.1 – Встановлення розширення Далі було встановлено підключення до сервера MongoDB.

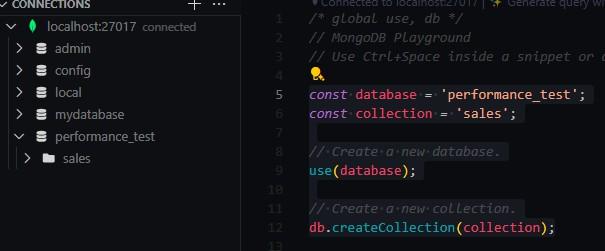
Рисунок 7.2 – Підключення встановлено Наступним кроком було створення колекції sales у бд performance\_test.

Рисунок 7.3 – Створення нової колекції

Далі було створено та запущено скрипт на Python для генерації та вставки 100000 документів до колекції sales.

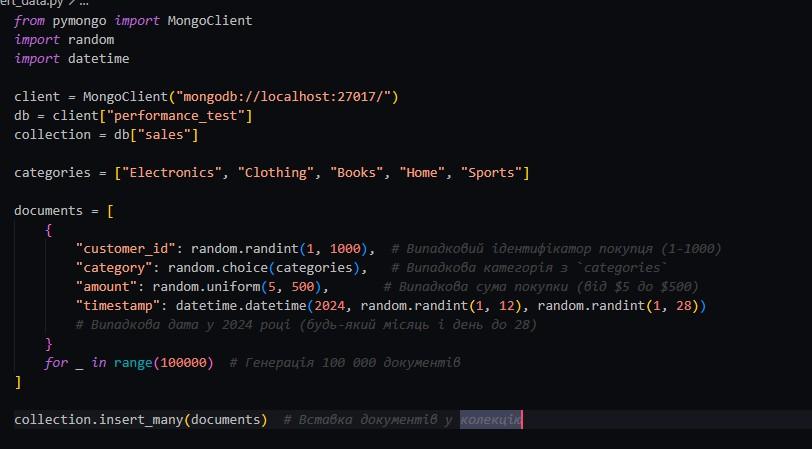


Рисунок 7.4 – Скрипт для генерації документів

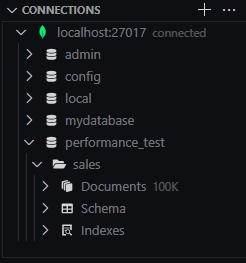


Рисунок 7.5 – У колекції sales 100000 документів

Далі було створено скрипт query\_no\_index.py що вимірює час запиту для отримання всіх документів sales, де значення поля Category – Electronics.

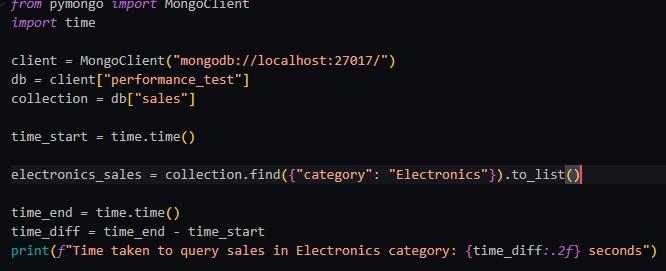


Рисунок 7.6 – Код скрипту



Рисунок 7.7 – Результат виконання

Після цього було створено та запущено скрипт create\_index.py для створення індексі для категорій та знову запущено query\_no\_index.py для порівняння

швидкості.



Рисунок 7.8 – Код create\_index.py



Рисунок 7.9 – Швидкість виконання запиту після додавання індексу Індексація прискорила роботу запиту приблизно на 27%.

Далі було створено складений індекс (category, timestamp) з використанням MongoDB Compass. Після цього було виконано запит {“category”:

“Electronics”} та переглянуто вкладку Explain.

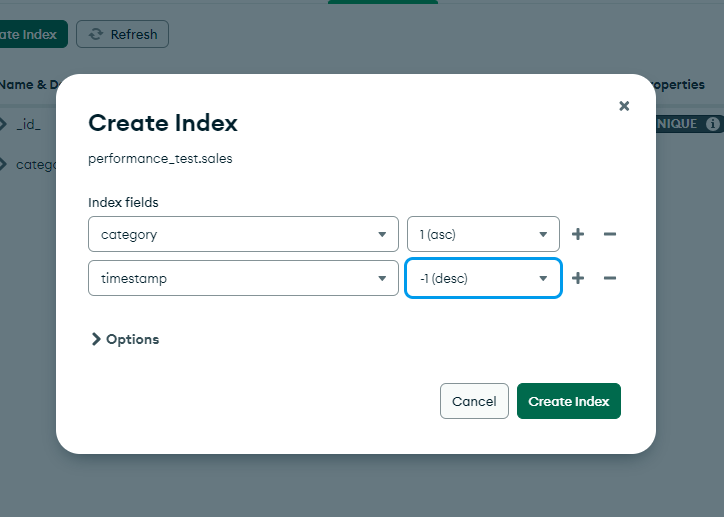


Рисунок 7.10 - Створення складеного індексу у MongoDB Compass

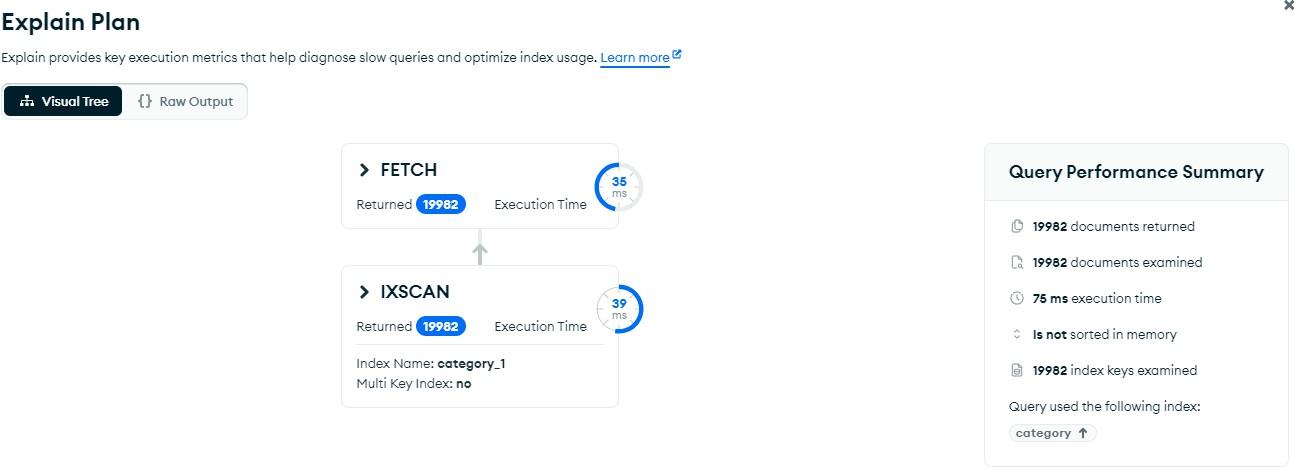


Рисунок 7.11 – Вкладка Explain. Використовується попередньо створений індекс category\_1

Як бачимо, новий, складений індекс не було застосовано. Щоб використовувався саме він можна видалити попередній.

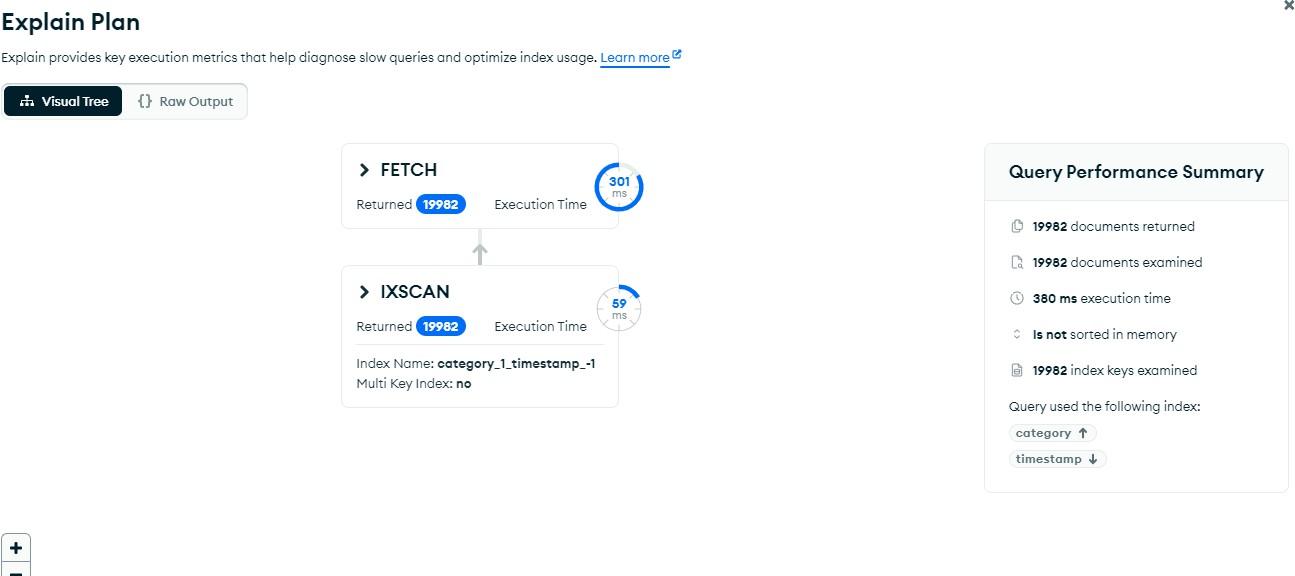


Рисунок 7.12 – Після видалення старого індекса застосовується складений

# Частина 2: Налаштування реплікації

Для початку роботи було запущено три екземпляри MongoDB в різних терміналах з параметром --replSet. Далі було виконано підключення до 1шого з них за допомогою MongoDB Compass, після чого виконано команди ініціалізації реплікації та додавання інших екземплярів.



Рисунок 7.13 – Команди налаштування реплікації

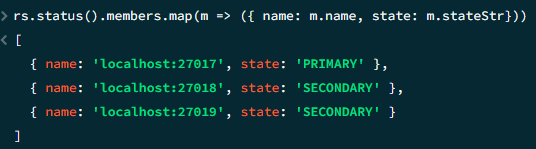


Рисунок 7.14 – Перевірка роботи кластеру реплікації

Після відключення екземпляра зі статусом “PRIMARY”, цей статус переходить до іншого.

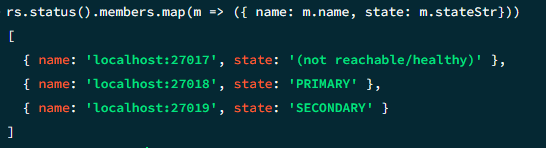


Рисунок 7.15 – Оновлений статус

# Частина 3: Реалізація шардінгу

Для початку роботи було запущено ще 2 екземпляри MongoDB:

mongod --shardsvr --port 27021 --dbpath D:/data/shard1 --bind\_ip localhost --replSet shardSet1

mongod --shardsvr --port 27022 --dbpath D:/data/shard2 --bind\_ip localhost --replSet shardSet2

Після цього до кожного з них було підключено mongosh та виконано rs.initiate()



Рисунок 7.16 – Команда rs.initiate()

Наступний крок – ініціалізація сервера конфігурації mongos. Для цього завдання було запущено ще один екземпляр mongodb, ініціалізовано його та запущено mongos:

mongod --configsvr --replSet configReplSet --port 27019 --dbpath D:\data\config1 --bind\_ip localhost

mongos --configdb configReplSet/localhost:27019 --bind\_ip localhost --

port 27016

Після цього було виконано підключення mongosh до серверу mongos та додано «шарди».

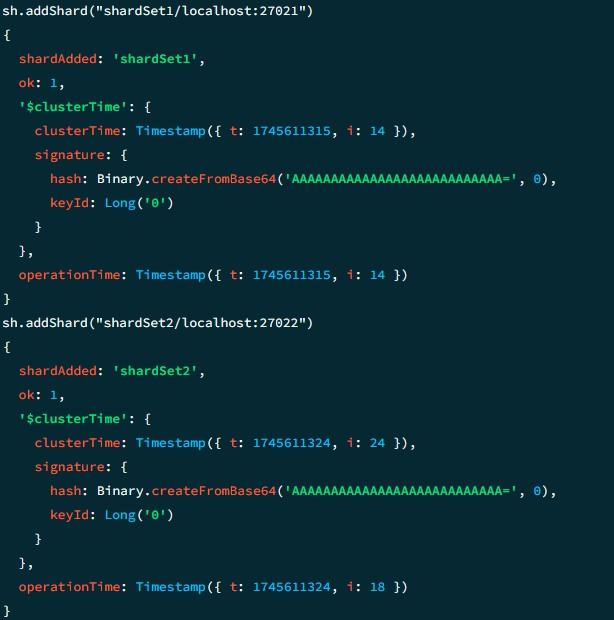


Рисунок 7.17 – Додавання «шардів»

Далі було налаштовано «шардинг» для колекції performance\_test.sales.

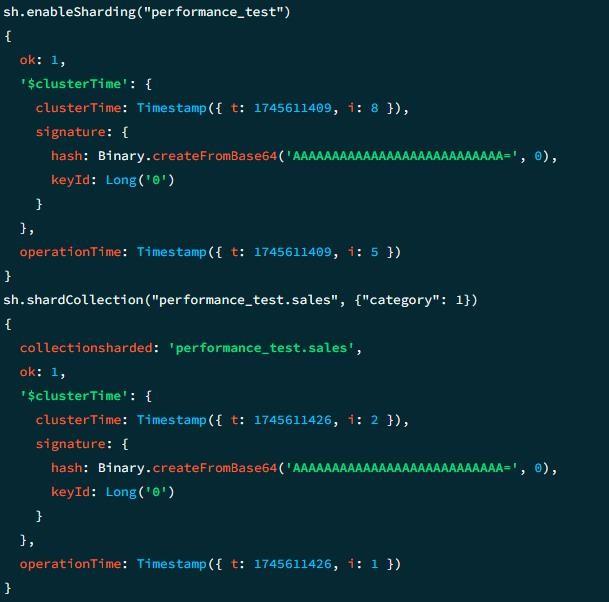


Рисунок 7.18 – Налаштування «шардингу» для колекції

Для перевірки «шардингу» було виконано команду sh.status().

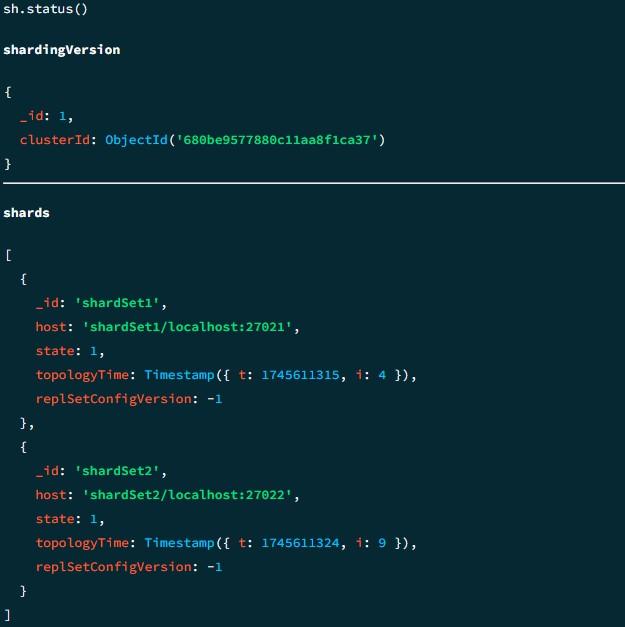


Рисунок 7.19 – Частина виводу sh.status()

# Додаткове завдання: візуалізація даних

Для візуалізації даних було створено новий скрипт sales\_per\_category\_chart.py, де було використано бібліотеки pymongo та matplotlib для підключення до бд та візуалізації даних відповідно.

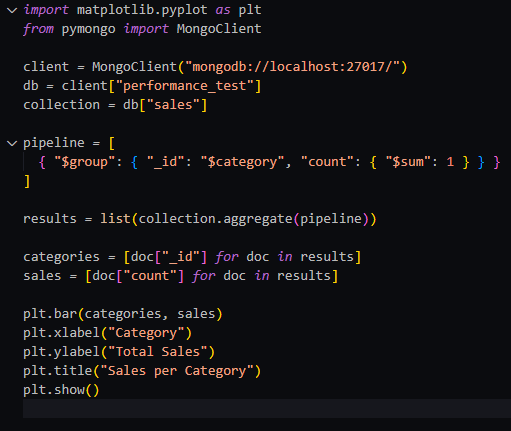


Рисунок 7.20 – Код скрипта

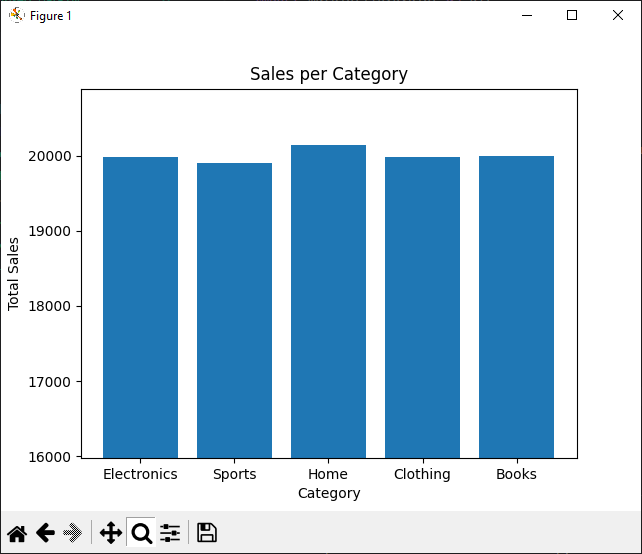


Рисунок 7.21 – Результат запуску програми

**Висновок:** під час роботи було розглянуто індексацію у MongoDB, реплікації та шардингу. Індексація – це один із способів оптимізації MongoDB для прискорення запитів. А реплікація, як засіб маштабування, покращує доступність даних та розподіл навантаження. У свою чергу, шардінг допомагає ефективно управляти великими обсягами даних, розподіляючи їх між вузлами.