**Міністерство освіти і науки України**

**Харківський національний університет ім. В. Н. Каразіна**

Факультет комп’ютерних наук

Кафедра

ЗАТВЕРДЖЕНО

Завідувач кафедри

\_\_\_\_\_\_\_\_

**Звіт**

про виробничу практику

| Оцінка \_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Члени комісії:  \_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) (прізвище, ініціали)  \_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) (прізвище, ініціали)  \_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) (прізвище, ініціали)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ р. | Виконав: студент 4 курсу, групи КС41  спеціальності: 122 «Комп’ютерні науки»  \_\_\_\_\_\_\_\_\_\_\_    Моцний Д.В.  (підпис) (прізвище, ініціали)  Керівник від бази практики  ст. викладач кафедри МСіТ  Дубинка А.М.  (ступень, звання, прізвище та ініціали)  Рекомендована оцінка: \_\_\_\_\_\_\_\_ балів  \_\_\_\_\_\_\_\_\_\_\_\_   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) (прізвище, ініціали) |
| --- | --- |

Харків – 2022

**ЗМІСТ**

[**ВСТУП**](#_heading=h.hpckz3s96p70) **2**

[Програма виробничої практики. Індивідуальне завдання](#_heading=h.g012l9rxrp25) 3

[**ОСНОВНА ЧАСТИНА**](#_heading=h.qg2bkauc02di) **4**

[Перелік джерел з анотаціями головних положень](#_heading=h.o02bh56qn7bq) 4

[**ВИСНОВОК**](#_heading=h.i5ppv0u5905w) **12**

[**ВИКОРИСТАНІ ДЖЕРЕЛА**](#_heading=h.ku2yh7ejwwnz) **12**

# ВСТУП

Виробнича практика проходила на базі Харківського національного університету імені В. Н. Каразіна, факультету комп’ютерних наук, кафедри моделювання систем і технологій.

Термін виробничої практики складав 9 тижнів (з 10 жовтня по 10 грудня 2022 року).

**Тема виробничої практики**: «Методи оптимізації запитів до NoSQL баз даних».

**Завдання на термін проходження практики**

Ознайомитися з предметною галуззю «Методи оптимізації запитів до NoSQL баз даних». Провести розширений пошук інформаційних джерел з цієї тематики, проаналізувати їх, зробити перелік джерел з анотаціями головних положень.

## Програма виробничої практики. Індивідуальне завдання

| **Пор.** | **Назви робіт, що має виконати студент** | **Тижні проходження практики** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|  | Отримання завдання. Обговорення постановки задачі | **+** | **+** | **+** |  |  |  |  |  |  |
|  | Знайомство з літературою та інтернет джерелами предметної галузі |  | **+** | **+** | **+** |  |  |  |  |  |
|  | Пошук інформаційних джерел, які описують методи оптимізації запитів до NoSQL баз даних |  | **+** | **+** | **+** | **+** | **+** | **+** | **+** |  |
|  | Аналіз знайдених результатів |  |  |  |  |  | **+** | **+** | **+** |  |
|  | Створення анотованого огляду знайдених інформаційних джерел |  |  |  |  |  |  | **+** | **+** | **+** |
|  | Підготовка звіту з виробничої практики. Підготовка презентації звіту |  |  |  |  |  |  |  | **+** | **+** |

**Індивідуальне завдання** було представлене сукупністю наступних задач:

* Знайомство з літературою та інтернет джерелами.
* Пошук інформаційних джерел, які описують методи оптимізації запитів до NoSQL баз даних.
* Аналіз знайдених результатів.
* Створення анотованого огляду знайдених інформаційних джерел.
* Підготовка звіту з виробничої практики.

# ОСНОВНА ЧАСТИНА

Початковим завданням було ознайомитися з предметною галуззю та зібрати інформаційні джерела в електронному вигляді.

Основним завданням було провести розширений пошук інформаційних джерел у мережі Інтернет, які стосуються предметної галузі «Методи оптимізації запитів до NoSQL баз даних». Після виконання цього завдання було створено архів з електронними версіями публікацій та статей, який включає в себе перелік статей.

Останнім кроком було створення загального переліку обраних джерел з анотаціями головних положень українською мовою.

## Перелік джерел з анотаціями головних положень

* Keshav Murthy. Approaches to Query Optimization in NoSQL. 2019. URL: <https://www.couchbase.com/blog/query-optimization-in-nosql-couchbase-mongodb/> (дата звернення: 20.11.2022).

Стаття описує можливості опитимізації запитиів до NoSQL баз даних на прикладі MongoDB, за допомогою таких способів як: створення коректних індексів, види Join, їх перегрупування. Перезапис запиту на основі евристики та складності виконання запиту. Особливо детально описано питання чому не можна створити кілька текстових індексів у MongoDB.

* Aristeidis Karras, Christos Karras, Dimitrios Samoladas, Konstantinos C. Giotopoulos, Spyros Sioutas. Query Optimization in NoSQL Databases Using an Enhanced Localized R-tree Index. 2022. URL: <https://link.springer.com/chapter/10.1007/978-3-031-21047-1_33> (дата звернення: 21.11.2022).

У цій роботі представлений покажчик просторових даних, тобто координат на площині або на карті. Цей індекс базується на R-дереві, яке підходить для просторових даних і розподіляється таким чином, щоб він міг масштабуватися та адаптуватися до величезних обсягів даних без втрати продуктивності. Оскільки розмір даних у сучасних програмах збільшується через різні джерела, типи та мультимодальні записи в базах даних, виникає потреба оптимізувати пошукові операції. Таким чином, індекси можна використовувати для вирішення проблеми швидкого зростання даних, оскільки вони підвищують продуктивність бази даних і згодом хмарного сервера, де вони зберігаються. Результати запропонованого методу є в усіх експериментах, і майбутні напрямки цієї роботи включають експерименти з різними даними.

* Divya Mahajan, Cody Blakeney, Ziliang Zong. Improving the energy efficiency of relational and NoSQL databases via query optimizations. 2019. URL: <https://www.sciencedirect.com/science/article/abs/pii/S2210537918301112> (дата звернення: 21.11.2022).

Оскільки великі дані стають нормою різноманітних промислових додатків, складність робочих навантажень бази даних і проектування системи баз даних значно зросла. Щоб вирішити ці проблеми, звичайні реляційні бази даних постійно вдосконалювалися, а бази даних NoSQL, такі як MongoDB і Cassandra, були запропоновані та впроваджені, щоб конкурувати з базами даних SQL. На додаток до традиційних показників, таких як час відповіді, пропускна здатність і ємність, сучасні системи баз даних висувають вищі вимоги до енергоефективності через великий обсяг даних, які потрібно зберігати, запитувати, оновлювати та аналізувати. Хоча десятиліття досліджень у спільнотах баз даних і обробки даних створили велику кількість літератури щодо оптимізації продуктивності, дослідження оптимізації енергоефективності історично залишалися поза увагою, і лише кілька досліджень досліджували енергоефективність систем баз даних. У цій статті досліджено серію методів оптимізації запитів для підвищення енергоефективності реляційних баз даних і баз даних NoSQL. У експериментах використовується загальноприйнятні тести (наприклад, Yahoo! Cloud Server Benchmark) і спеціальні набори даних (конвертовані з ~100 ГБ даних Twitter), щоб оцінити ефективність різних методів оптимізації. Проводиться перехресний аналіз баз даних реляційної бази даних (MySQL) і баз даних на базі NoSQL (MongoDB і Cassandra), щоб порівняти їх продуктивність і енергоефективність. Крім того, розлядаються різноманітні методи оптимізації, які можуть покращити енергоефективність без шкоди для продуктивності баз даних, отриманих із даних Twitter. Крім того, досліджується вплив динамічного масштабування напруги та частоти (DVFS) на продуктивність та енергоефективність MySQL, MongoDB і Cassandra.

* Reynan Anislag, Jay Rico. NoSQL: An Optimization Approach in a Stand-Alone System. 2021. URL: <https://asianjournal.org/online/index.php/ajms/article/view/442> (дата звернення: 21.11.2022).

З огляду на деякі обмежувальні фактори баз даних SQL, NoSQL було вивчено і зараз є популярною базою даних, яка використовується найбільшими компаніями світу, такими як Facebook і Google, для роботи з неструктурованими великими даними, які вони збирають і керують щодня. У автономній програмі, як-от система нарахування заробітної плати, використання баз даних SQL також стає проблемою, особливо коли зібрані дані швидко зростають. Нормалізація або оптимізація запитів має невелику перевагу, тоді як NoSQL вирішив цю проблему набагато краще. У новій моделі база даних NoSQL (MongoDB) була розроблена для роботи з транзакційними даними, зокрема під час фази читання, які повинні бути взяті з транзакційної таблиці. З іншого боку, база даних SQL (MySQL) все ще працює з іншими таблицями, які мають стабільні атрибути. Тест проводився з використанням автономної системи над звичайним застосуванням баз даних NoSQL, таких як розподілені онлайн-системи. Комбіновані бази даних SQL і NoSQL показали значні результати з точки зору продуктивності, ніж лише база даних SQL. Очевидно, що NoSQL добре працює з SQL. NoSQL був би одним із найкращих варіантів для оптимізації продуктивності системи. Це не тільки ідеально працює з неструктурованими даними в розподілених системах, але також має велику перевагу при застосуванні до автономних систем, особливо коли він збирає та керує великим обсягом даних.

* Jay Patel. Guidelines for Modeling and Optimizing NoSQL Databases. 2012. URL: <http://www.ebaytechblog.com/2012/07/16/cassandra-data-modeling-best-practices-part-1/> (дата звернення: 21.11.2022).

У цій статті розповідається, про моделювання даних за допомогою Cassandra, як було обрано використання стовпців і сімейств стовпців і оптимізації запитів. В цій статті, застосовано високорівневий підхід Джея та його команди. Висновки з цього підходу до моделювання даних та оптимізації запитів, можна застосувати до будь-якої бази даних NoSQL, включаючи Cassandra, MongoDB, Redis та інші.

* Christine Niyizamwiyitira, Lars Lundberg. PERFORMANCE EVALUATION OF SQL AND NOSQL DATABASE MANAGEMENT SYSTEMS IN A CLUSTER. 2017. URL: <https://aircconline.com/ijdms/V9N6/9617ijdms01.pdf> (дата звернення: 23.11.2022).

У цьому дослідженні оцінюється продуктивність систем керування базами даних SQL і NoSQL, а саме; Cassandra, CouchDB, MongoDB, PostgreSQL і RethinkDB. Використано кластер із чотирьох вузлів для запуску системи баз даних із зовнішніми генераторами навантаження. Оцінка проводиться з використанням даних Telenor Sverige, телекомунікаційна компанія, яка працює у Швеції. Експерименти проводяться з використанням три набори даних різного розміру. Пропускна здатність запису та затримка, а також пропускна здатність читання та затримка оцінюється для чотирьох запитів; а саме запит відстані, запит k-найближчого сусіда, запит діапазону та запит регіону. Для операцій запису Cassandra має найвищу пропускну здатність, коли використовується кілька вузлів, тоді як PostgreSQL має найнижчу затримку та найвищу пропускну здатність для одного вузла. Для читання

операції MongoDB має найнижчу затримку для всіх запитів. Проте найвищий у Кассандри пропускна здатність для читання. Пропускна здатність зменшується, оскільки розмір набору даних збільшується як для запису, так і для читання як послідовний, так і довільний доступ. Однак це зниження є більш значним для випадкових читати і писати. У цьому дослідженні представлено досвід, з різними базами даних.

* Jiao Dai. SQL to NoSQL: What to do and How. 2019. URL: <https://iopscience.iop.org/article/10.1088/1755-1315/234/1/012080/pdf> (дата звернення: 23.11.2022).

Через складну логіку управління підприємством корпоративні програми можуть містити тисячі таблиць із масовими зв’язками чи обмеженнями між ними. Традиційні підприємства все ще використовують RDBMS, тоді як система NoSQL широко поширена завдяки своїй чудовій продуктивності та високій доступності. Таким чином, інструменти для міграції з РСУБД на базу даних NoSQL дуже доступні. Однак схема має значення не лише в RDBMS, але й у системі NoSQL для кращої відповіді на запити. У цій статті описано технологію перетворення бази даних, включаючи перетворення схем, переклад запитів і оптимізацію запитів.

* David Byrne. Speed up Your AWS DynamoDB Queries. 2021. URL: <https://medium.com/henngeblog/speed-up-your-aws-dynamodb-queries-455280e40226> (дата звернення: 27.11.2022).

У цій публікації розглянуто сценарій порівняльного аналізу, який був написаний, щоб визначити ефективність покращення продуктивності запитів DDB. Цей тестовий сценарій просто виконував запит, згаданий вище, і визначав, скільки часу тривала операція, використовуючи модуль timeit Python.

* Lutfi Rahmatuti Maghfiroh. Study of Search Algorithm Optimization from Multi-Version Data Warehouse using NoSQL Non-relational Database. 2022. URL: <https://proceedings.stis.ac.id/icdsos/article/view/74> (дата звернення: 28.11.2022).

Статистиці Індонезії, яка виробляє великі дані, потрібне ефективне та оптимальне зберігання. Дослідження, пов’язані з багатоверсійним сховищем даних (MVDW), яке використовує сам NoSQL на основі документів, намагалися розробити для збереження даних BPS і запропонували алгоритм для зберігання та пошуку даних. Ця стаття присвячена дослідженню методів оптимізації алгоритму для зменшення часу, який використовується в процесі зберігання та пошуку даних, коли це необхідно. Алгоритм, запропонований у цьому документі, фокусується на процесі зберігання даних, пропонуючи модель зберігання, яка узагальнює кодування змінних у сховищі даних, що використовується, щоб подальший пошук даних можна було здійснювати легше й оптимальніше. Інші методи оптимізації також здійснюються шляхом застосування методів оптимізації запитів для підтримки та покращення оптимізації запропонованого алгоритму. Результати двох проведених методів оптимізації можна назвати успішними, оскільки час, витрачений на процес пошуку даних за допомогою алгоритму після застосування методу оптимізації, було зменшено порівняно з процесом пошуку даних з використанням алгоритмів, які були розроблені попередніми дослідженнями.

* Ali Ben Ammar. QUERY OPTIMIZATION TECHNIQUES IN GRAPH DATABASES. 2016. URL: <https://www.academia.edu/28363536/QUERY_OPTIMIZATION_TECHNIQUES_IN_GRAPH_DATABASES> (дата звернення: 29.11.2022).

Графові бази даних (GDB) виникли, щоб подолати обмеження традиційних баз даних для зберігання та керування даними з графоподібною структурою. Сьогодні вони є вимогою для багатьох програм, які керують графоподібними даними, як-от соціальні мережі. Більшість методів, застосованих для оптимізації запитів у графових базах даних, використовувалися в традиційних базах даних, системах розповсюдження або вони були натхненні теорією графів. Однак їхнє повторне використання в графових базах даних має враховувати основні характеристики графових баз даних, такі як динамічна структура, тісно взаємозв’язані дані та можливість ефективного доступу до зв’язків даних. У цій статті розглянуто методи оптимізації запитів у базах графічних даних. Зокрема, зосереджено увагу на функціях, які вони запровадили для покращення запитів до даних у вигляді графіків.

# ВИСНОВОК

В результаті проходження виробничої практики ознайомився з предметною галуззю «Методи оптимізації запитів до NoSQL баз даних».

Було створено перелік інформаційних джерел, які були знайдені шляхом розширеного пошуку в мережі Інтернет та зроблено анотації головних положень знайдених статей.

# ВИКОРИСТАНІ ДЖЕРЕЛА

* Keshav Murthy. Approaches to Query Optimization in NoSQL. 2019. URL: <https://www.couchbase.com/blog/query-optimization-in-nosql-couchbase-mongodb/> (дата звернення: 20.11.2022).
* Aristeidis Karras, Christos Karras, Dimitrios Samoladas, Konstantinos C. Giotopoulos, Spyros Sioutas. Query Optimization in NoSQL Databases Using an Enhanced Localized R-tree Index. 2022. URL: <https://link.springer.com/chapter/10.1007/978-3-031-21047-1_33> (дата звернення: 21.11.2022).
* Divya Mahajan, Cody Blakeney, Ziliang Zong. Improving the energy efficiency of relational and NoSQL databases via query optimizations. 2019. URL: <https://www.sciencedirect.com/science/article/abs/pii/S2210537918301112> (дата звернення: 21.11.2022).
* Reynan Anislag, Jay Rico. NoSQL: An Optimization Approach in a Stand-Alone System. 2021. URL: <https://asianjournal.org/online/index.php/ajms/article/view/442> (дата звернення: 21.11.2022).
* Jay Patel. Guidelines for Modeling and Optimizing NoSQL Databases. 2012. URL: <http://www.ebaytechblog.com/2012/07/16/cassandra-data-modeling-best-practices-part-1/> (дата звернення: 21.11.2022).
* Christine Niyizamwiyitira, Lars Lundberg. PERFORMANCE EVALUATION OF SQL AND NOSQL DATABASE MANAGEMENT SYSTEMS IN A CLUSTER. 2017. URL: <https://aircconline.com/ijdms/V9N6/9617ijdms01.pdf> (дата звернення: 23.11.2022).
* Jiao Dai. SQL to NoSQL: What to do and How. 2019. URL: <https://iopscience.iop.org/article/10.1088/1755-1315/234/1/012080/pdf> (дата звернення: 23.11.2022).
* David Byrne. Speed up Your AWS DynamoDB Queries. 2021. URL: <https://medium.com/henngeblog/speed-up-your-aws-dynamodb-queries-455280e40226> (дата звернення: 27.11.2022).
* Lutfi Rahmatuti Maghfiroh. Study of Search Algorithm Optimization from Multi-Version Data Warehouse using NoSQL Non-relational Database. 2022. URL: <https://proceedings.stis.ac.id/icdsos/article/view/74> (дата звернення: 28.11.2022).
* Ali Ben Ammar. QUERY OPTIMIZATION TECHNIQUES IN GRAPH DATABASES. 2016. URL: <https://www.academia.edu/28363536/QUERY_OPTIMIZATION_TECHNIQUES_IN_GRAPH_DATABASES> (дата звернення: 29.11.2022).

Керівник практики

від кафедри \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище, ініціали) (підпис)