Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів оптимізації баз даних Програмних систем

(тема)

Виконав:
Випускник 2 курсу, групи ІПЗм-22-4
Кайдаш С.А.
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми Освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Керівник доц. Мар'їн А.С.
(посада, прізвище)

Допускається до захисту
Зав. кафедри _____ З.В. Дудар
(підпис) (прізвище, ініціали)

2024 р.
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва)

Тип програми освітньо-наукова програма
(освітньо-професійна або освітньо-наукова)

Освітня програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
« __ » _____ 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Кайдаша Станіслава Андрійовича
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів оптимізації баз даних програмних систем» затверджена наказом університету від 29.03. 2024р. № 250 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 19.06.2024

3. Вихідні дані до роботи: *методи оптимізації баз даних відомості про методи аналізу статистичних даних, використовувати СКБД MySQL та ОС Ubuntu, мову програмування PHP та фреймворк Laravel*

4. Перелік питань, що потрібно опрацювати в роботі *мета роботи, аналіз стану розв'язання проблеми і постановка задачі, огляд, аналіз та дослідження методів оптимізації баз даних програмних систем за часом обробки sql-запиту, створення моделі з тестовим набором даних та можливістю його зміни*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі та постановка задачі	30.03 – 14.04.24	*виконано*
2	Аналіз та вибір API для дослідження	15.04 – 24.04.24	*виконано*
3	Аналіз та моделювання предметної області	25.04 – 28.04.24	*виконано*
4	Планування експериментів	29.04 – 08.05.24	*виконано*
5	Програмна реалізація кожного з обраних для дослідження API	09.05 – 19.05.24	*виконано*
6	Експериментальні дослідження	20.05 – 25.05.24	*виконано*
7	Аналіз результатів експериментальних досліджень та розробка рекомендацій	25.05 – 28.05.24	*виконано*
8	Написання та оформлення статті та тез доповіді	28.05 – 31.05.24	*виконано*
9	Підготовка пояснювальної записки	01.06 – 09.05.24	*виконано*
10	Підготовка презентації та доповіді	10.06 – 12.06.24	*виконано*
11	Нормоконтроль	13.06 – 14.06.24	*виконано*
12	Рецензування	14.06 – 15.06.24	*виконано*
13	Занесення диплома в електронний архів	16.06.2024	*виконано*
14	Попередній захист	17.06.2024	*виконано*
15	Допуск до захисту у зав. кафедри	18.06.2024	*виконано*

Дата видачі завдання 30 березня 2024р.

Студент (ка)  Кайдаша С. А.

(підпис)

Керівник роботи доц. Мар'їн С.О.

(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ ABSTRACT

Кваліфікаційна робота магістра містить: 59 с., 7 мал, 3 табл., 11 джерел.

MYSQL, PHP, LARAVEL, PHPSTORM, АТЕСТАЦІЙНА РОБОТА, МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ, СТАТИСТИЧНЕ ДОСЛІДЖЕННЯ, ДИСПЕРСІЙНИЙ АНАЛІЗ, СИСТЕМА, КОМПОНЕНТ.

Об’єктом дослідження є база даних програмної системи. Предметом дослідження є методи оптимізації баз даних.

Метою роботи є порівняння методів оптимізації баз даних.

Методи розробки базуються на фреймворку Larevel мові програмування PHP.

Підчас дослідження здійснена програмна реалізація статистичного методу математичного моделювання. Було виділено та досліджено основні фактори впливу на поведінку системи.

MYSQL, PHP, LARAVEL, PHPSTORM, CERTIFICATION WORK, MATHEMATICAL MODELING, STATISTICAL RESEARCH, ANALYSIS OF VARIANCE, SYSTEM, COMPONENT.

The object of research is a database of a software system. The subject of research is methods of database optimization.

The purpose of the study is to compare database optimization methods.

The development methods are based on the Larevel framework of the PHP programming language.

In the course of the study, the software implementation of the statistical method of mathematical modeling was carried out. The main factors influencing the system behavior were identified and investigated.

Я, *Кайдаш Станіслав Андрійович,* студент групи ІПЗм-22-4, здобувач вищої

освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів оптимізації баз даних програмних систем», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElAr KhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений(а) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП

База даних, також відома як інформаційне сховище, — це набір записів, упорядкованих відповідно до певної концепції та надає специфіку характеристик і зв’язків між компонентами. Бази даних часто містять таблиці, збережені процедури, подання, тригери, індекси та інші компоненти. Система керування базами даних, іноді відома як СУБД, — це набір програмних засобів, які дозволяють користувачам взаємодіяти з різними компонентами бази даних [1].

З моменту створення перших баз даних програмісти час від часу стикалися з проблемою оптимізації коду для зниження використання ресурсів і підвищення продуктивності. Труднощі з максимізацією операційної продуктивності зберігаються, а іноді погіршуються через необхідність зберігати величезні обсяги даних, але технологія зберігання структурованої інформації прогресувала, оскільки кількість даних, які потрібно зберігати, зросла. Крім того, оскільки передбачалося, що системи керування базами даних працюють як стандартне програмне забезпечення та забороняють втручання у власне програмне забезпечення, вважається, що єдиний спосіб оптимізувати систему керування базами даних щодо швидкості – це використання стандартних інструментів, які постачаються користувачеві, а також їх комбінації. код. Отже, проблема оптимізації продуктивності обробки запитів на рівні системи управління базами даних (СУБД) стає все більш актуальною в міру розвитку технологій.

Основною метою цієї роботи є визначення оптимальної архітектури реляційної бази даних і змін конфігурації для певного типу програмної системи, щоб забезпечити швидшу обробку запитів. Найважливіші завдання, які необхідно виконати для дослідження, такі:

– розподіл аспектів, які впливають на швидкість обробки запитів до бази даних, і використання їх як коефіцієнті оптимізації;

– вивчення впливу цих компонентів та їх взаємодії на загальну продуктивність швидкості шляхом дослідження;

– введення експериментальних даних у математичну модель для узагальнення;

– генерування рекомендацій щодо вдосконалення бази даних, які можна використовувати для сценаріїв реального світу або майбутніх досліджень.

В даній роботи як об’єкт дослідження використовується звичайна структура бази даних. Оскільки бази даних часто використовуються в розробці онлайн-магазинів, оптимізація баз даних для таких систем може отримати користь від результатів дослідження.

Основна увага дослідження зосереджена на застосуванні раніше встановлених параметрів для оптимізації часу обробки запитів до бази даних на рівні СУБД. Щоб виявити чітку закономірність у темі залежності дослідження від відповідних факторів впливу, можна використати набір експериментів і математичне моделювання для збору й узагальнення набору експериментальних даних.

Результати дослідження відображаються у вигляді числових даних, які добре читаються, і підкреслюють важливість кожного компонента у впливі на кінцевий результат з часом. Ця інформація може слугувати основою для майбутніх досліджень у відповідній темі та надати вказівки для створення баз даних програмних систем, подібних до тієї, що вивчається.

1 Аналіз предметної галузі

1.1 АНАЛІЗ СТАНУ РОЗВ’ЯЗАННЯ ПРОБЛЕМИ

З розвитком технологій і зростанням вимог до обробки та зберігання даних швидкість цих систем стає все більш вирішальним фактором. За останні десять років розмір промислових дисків збільшився через необхідність зберігати величезні обсяги даних. Але потреба в більш досконалих системах обробки зросла разом із обсягом зберігання.

З точки зору користувача, система повинна діяти послідовно незалежно від того, скільки даних вона обробляє. Системи обробки та зберігання даних зазвичай визначаються як системи, які працюють переважно з набором структурованих даних, які зазвичай зберігаються в кількох базах даних. Найпоширенішим типом бази даних є реляційна база даних, яка зберігає інформацію в таблицях. Тому кожна таблиця в цій концепції бази даних пов’язана з окремим об’єктом. Цей тип становить близько 80% світового ринку, хоча проблеми починають виникати через використання реляційних типів для вирішення проблеми зберігання величезних обсягів даних. Реляційні бази даних мають дві ключові переваги: вони зберігають структуровані дані та мають архітектурні особливості, які захищають цілісність даних. Зв’язки між компонентами даних забезпечують цю функціональність; однак через витрати, пов’язані з підтримкою та перевіркою цих посилань, обробка великих обсягів даних і даних із поганим форматом може бути складною.

Виконання команди INSERT для запису в MySQL, який містить понад 350 тисяч рядків, може зайняти до 6 хвилин і 21 секунди, виходячи з досліджень, проведених на системах з величезними наборами даних (понад 500 мільйонів записів), представлених Oracle. Бази даних з більш ніж 2 мільйонами записів і до 4 хвилин і 10 секунд для виконання запиту на читання за допомогою команди SELECT [6]. Ці вимірювання вказують на те, що використовувати великомасштабні бази даних на основі SQL стає все важче, і що цим базам даних може знадобитися оптимізувати продуктивність обробки запитів.

Немає чітких рекомендацій щодо прискорення СУБД, оскільки кожна база

даних відрізняється і залежить від контексту, в якому вона використовується; незважаючи на це, думки професіоналів щодо методів, які використовуються, коли підходи адаптовані для конкретної програми бази даних, є послідовними.

Цей список може включати:

- додавання індексу до кількох стовпців таблиці;
- включання дублюючих стовпців;
- розділення столів як по горизонталі, так і по вертикалі;
- денормалізація баз даних

Як би дивно це не здавалося, стає все важче знайти результати дослідження, які демонструють ефективність будь-якої з перерахованих вище стратегій оптимізації в типовій ситуації. Підсумок досліджень у відкритому доступі може привести до висновку, що не існує послідовних вказівок щодо оцінки ефективності будь-якого методу взагалі. Це може бути так, оскільки успішність подібних методів оптимізації значно відрізняється в залежності від середовища, в якому вони застосовуються, на основі різноманітних обставин. До них відносяться кількість рядків у кожній таблиці, розмір таблиць, типи даних окремих стовпців і структура бази даних. Тим не менш, потреба в більш глибокому дослідженні методів оптимізації бази даних для більш спеціалізованих програм баз даних не зникає одночасно. Таке дослідження може бути дуже корисним, оскільки його можна реально використовувати для створення баз даних для традиційних програмних систем. Тому було вирішено зосередитися на конкретному контексті використання бази даних у цьому дослідженні.

1.2 Постановка задачи

Об’єкт і предмет дослідження

Об’єктом дослідження є база даних програмної системи, яка використовується для обробки та зберігання великих обсягів даних. Предметом дослідження є методи оптимізації баз даних, спрямовані на підвищення їх продуктивності та

ефективності в умовах обробки великих масивів даних.

Мета дослідження

Метою роботи є порівняння різних методів оптимізації баз даних для визначення найефективніших підходів, що дозволяють покращити продуктивність систем зберігання та обробки даних, зокрема реляційних баз даних, в умовах сучасних вимог до швидкості і обсягів оброблюваної інформації.

Завдання дослідження

1. **Аналіз існуючих методів оптимізації**: Вивчення різних методів оптимізації баз даних, таких як додавання індексів, дублювання стовпців, розділення таблиць по горизонталі та вертикалі, і денормалізація баз даних.

2. **Визначення основних факторів впливу**: Ідентифікація ключових факторів, що впливають на продуктивність систем зберігання та обробки даних, включаючи кількість рядків у таблицях, розмір таблиць, типи даних у стовпцях і структура бази даних.

3. **Програмна реалізація методів оптимізації**: Реалізація та тестування методів оптимізації на основі фреймворку Laravel і мови програмування PHP. Використання інтегрованого середовища розробки PHPStorm для розробки та відлагодження програмного коду.

4. **Статистичний аналіз ефективності методів оптимізації**: Використання методів математичного моделювання та статистичного аналізу, зокрема аналізу варіації, для оцінки ефективності різних методів оптимізації в різних умовах експлуатації баз даних.

5. **Порівняння результатів досліджень**: Проведення порівняльного аналізу отриманих результатів, формулювання висновків і рекомендацій щодо вибору оптимальних методів оптимізації для різних сценаріїв використання баз даних.

Методи дослідження

Математичне моделювання:

Найчастіше виділяють три типи моделей:

1. логіко-математичні;
2. істотно-математичні;
3. Фізичні.

Інструменти

1. **MySQL**: Основна система управління базами даних (СУБД), що використовується для тестування методів оптимізації.
2. **PHP**: Мова програмування, яка використовується для реалізації алгоритмів та програмних модулів.
3. **Laravel**: Фреймворк для розробки веб-додатків і роботи з базами даних, що забезпечує зручний інструментарій для оптимізації.
4. **PHPStorm**: Інтегроване середовище розробки (IDE), що використовується для написання, відлагодження та тестування програмного коду.

2 ВИБІР МЕТОДІВ ДОСЛІДЖЕННЯ

2.1 Математичне моделювання

У дослідженнях використовуються математичні моделі систем, а математичне моделювання є засобом вирішення питання синтезу або аналізу певної складної системи. Інші значення «математичного моделювання» включають процес створення та вдосконалення математичної моделі системи для вивчення та відтворення деяких її характеристик, а також застосування математичних методів для передачі результатів дослідження назад до реальної системи. У наукових публікаціях цей термін часто використовується більш вузько. «Математичне моделювання» в цьому сенсі стосується виключно процесу створення або аналізу моделей.

Об’єкт, створений навмисно для імітації досліджуваного об’єкта або процесу, називається моделлю. Якщо безпосереднє вивчення предмета неможливе або це ускладнюється матеріально-технічними чи концептуальними проблемами, він відтворює структуру та властивості предмета в обсязі, необхідному для його вивчення. Створення моделі полегшує аналіз чогось. Об'єкт або процес, для якого створюється модель, називається прототипом.

Найчастіше виділяють три типи моделей:

– логіко-математичні;

– істотно-математичні;

– фізичні.

Прототипи та принципово математичні моделі математично описуються ідентичним способом, незважаючи на можливі фізичні відмінності.

У логіко-математиці моделі будують із символів. Ці моделі є лише обчислювальними та абстрактними. Важко розрізнити різні форми логічних і фундаментальних математичних моделей.

Фізичні моделі подібні досліджуваному об'єкту в природі. У деяких випадках розмір, вміст або інші фізичні характеристики такої моделі можуть просто відрізнятися від оригіналу, що не матиме жодного впливу на результати дослідження.

Перед початком процесу моделювання необхідно виконати кілька попередніх умов, але чітких правил вибору та побудови моделей немає:

– ключові елементи прототипу повинні бути відображені в моделі;

– щоб підтримати дослідження, модель має бути достатньою, тобто вона має точно відображати основні характеристики прототипу;

– лінійність, детермінізм, динамізм і дискретність прототипу повинні бути враховані при проектуванні.

2.2 Класифікація моделей та стадії моделювання

За основними характеристиками виділяють наступні категорії моделей:

– детерміновані та імовірнісні;

– статичні та динамічні;

– лінійні та нелінійні;

– безперервні та дискретні.

Модель можна вважати безперервною, якщо часові зміни, які вона зазнає, є безперервними. Якщо зміни моделі мають дискретні значення, то модель слід вважати дискретною.

Якщо концепція моделі враховує час, то її можна назвати динамічною. Інакше він буде статичним.

Якщо модель містить випадкові змінні або їхні властивості як ознаки, тоді її слід класифікувати як імовірнісну. У детермінованих моделях немає потенційних компонентів.

Модель можна віднести до лінійної, якщо її реакція на зміни зовнішнього середовища пропорційна величині цих змін; якщо ні, він буде класифікований як нелінійний.

Нижче наведено фази зазначеного процесу моделювання:

– побудова абстрактної моделі з використанням аналізу інформації та попередніх гіпотез;

– створення реальної (цифрової) моделі на основі спостережень і результатів експерименту.

Розробка абстрактної моделі вимагає наступних кроків:

– поділ важкої роботи або питання на менші, більш зрозумілі;

– пошук аналогів;

– внесення символів;

– запис цільової функції та обмежень

Складне завдання (проблему) краще розбити на менші, легші для розуміння компоненти. Наприклад, спочатку розробіть окремі локальні моделі, а потім об’єднайте їх у єдину глобальну модель. Знаходження аналога або зв’язку між математичним описом події, процесу чи об’єкта дослідження та відомим типом моделі є одним із фундаментальних елементів моделювання. Обмеження математичного опису певним типом суттєво спрощує процес розробки моделі та проведення додаткового аналізу. Проте регулярність слід використовувати з обережністю під час пошуку паралелей. Пошук аналога — одна з менш формалізованих частин дослідження, яка потребує відповідних навичок і знань. Однією з частин моделювання є процес введення символьного запису. Математичною моделлю є кодифікований запис або запис, представлений символами. Після завершення абстрактного моделювання змінні, що використовуються для визначення цільової функції та обмежень, зазвичай виражаються математично.

Процес побудови конкретної моделі передбачає:

– виявлення зв’язків між вхідними та вихідними параметрами об’єкта, явища чи процесу;

– вивчення ефектів окремих параметрів і базових змінних;

– встановлення обмежень у певному форматі;

– вдосконалення та оцінка розробленої моделі.

Успішність і практична користь процесу моделювання визначаються тим, наскільки сформовані явні залежності між вхідними і вихідними параметрами, що відображають глибинні закономірності досліджуваного явища, процесу або об'єкта. Якщо виявлені залежності не відповідають справжнім зв’язкам, тоді всі майбутні результати, створені моделлю, будуть хибними та позбавленими реальної сутності. Математичне моделювання стає надзвичайно корисним для

навчання надскладних процесів [2].

Процес підготовки та виконання математичного моделювання зазвичай включає такі кроки:

– формулювання, яке передбачає вибір набору засобів репрезентації (символів, візуальних образів тощо) для абстрактних ідей;

– маніпулювання, яке передбачає зміну символічного уявлення з метою прогнозування результату подібних маніпуляцій у реальному світі;

– абстракція, яка є практикою ігнорування відмінностей між речами та явищами матеріального світу на користь характеристик, які багато з них мають спільні;

– аналіз, використання математичних висновків шляхом їх інтерпретації в практичному контексті;

– аксіоматизація, формулювання тих якостей, виведених із реальності, які часто використовуються при маніпулюванні абстрактними символами, які представляють як матеріальний світ, так і реальність.

Розрізнивши макро- та мікрофази проектування, системний підхід може бути використаний для визначення точного порядку створення моделей. Будується модель зовнішнього середовища, визначаються ресурси та обмеження для розробки моделі системи, а модель системи та стандарти для оцінки придатності моделі M для фактичної системи S вибираються під час фази макропроектування. Для всього цього використовуються дані щодо фактичної системи S та зовнішнього середовища E.

Для того, щоб модель могла відтворювати певні характеристики продуктивності реальної системи S, оптимальна стратегія управління вибирається на основі продуктивності системи протягом фази моделювання, після створення моделей системи та зовнішнього середовища.

2.3 Принципи моделювання

Модель — штучно створений зразок, який у простішому та грубішому вигляді нагадує досліджуваний об’єкт (явище) шляхом відображення та

відтворення його структури, властивостей, зв’язків і взаємозв’язків між його елементами. Він може мати форму схеми, фізичних конструкцій, знакових форм або формул [3].

Принципи моделювання :

– принцип адекватності інформації: без будь-яких попередніх знань про річ побудова моделі неможлива. Моделювання безглузде, коли доступна вся інформація. Можна створити модель системи, коли досягнуто певного ступеня інформаційної достатності;

– принцип доцільності: створена модель має сприяти вчасному виконанню поставлених завдань дослідження;

– принцип множинної моделі: жодна модель не показує всю систему; лише його частина. Для всебічного аналізу необхідно побудувати кілька моделей досліджуваного процесу, і кожна нова модель повинна уточнювати попередні;

– принцип систематичності: набір взаємопов'язаних підсистем використовується для моделювання досліджуваної системи з використанням стандартних математичних підходів. Тим не менш, характеристики системи не збігаються з сукупними характеристиками її компонентів;

– принцип параметризації: Певні підсистеми модельованої системи, такі як вектор, матриця, графік або формула, можуть бути описані одним параметром.

Для більш повного виконання цієї роботи математична модель повинна задовольняти наступним вимогам.

Адекватним відображенням мети завдання системної моделі є те, що він послідовний. Але це не сувора відповідність, просто стільки, скільки потрібно (кількісний ступінь, кількість маєтків тощо).

Реалізованість — це відсутність суперечностей і підпорядкованість математичної основи моделі всім загальним математичним принципам, а також фізична життєздатність.

Інформативність - це здатність знаходити інформацію, необхідну для розуміння особливостей теми, що цікавить.

Чутливість — це здатність розпізнавати зміни в характеристиках стану у відповідь на зміни фундаментальних параметрів системи та зовнішніх факторів.

Еволюційний відноситься до здатності моделі розвиватися та адаптуватися у відповідь на зміну умов завдання та вимог до результатів.

Часова інваріантність: здатність проводити дослідження протягом необхідного часу.

Універсальність — це здатність вивчати широкий спектр проблем і інтегрувати модель з моделями з інших рівнів ієрархії без значного збільшення складності моделі.

Якість відтворення кінцевого результату при застосуванні до моделі процесу та системи вимірюється точністю. Часто коротка, неточна відповідь є більш корисною, ніж більша, детальніша.

Ефективність з точки зору вартості: забезпечення адекватного рівня витрат на моделювання.

Ергономіка: простота та практичність урахування ергономіки роботи системи та здійснення зв'язку між дослідником і моделлю на етапі моделювання.

Декомпозиція: здатність використовувати моделі окремих підсистем як самостійні системи в наукових дослідженнях.

2.4 Дисперсійний аналіз та його різновиди

Щоб організувати випробування, які можна порівняти з поточним експериментом, і оцінити вплив різних факторів на результат експерименту, використовується статистичний підхід, відомий як дисперсійний аналіз. Цей процес дозволяє порівнювати кілька зразків, кожен з яких може мати різне значення для даного атрибута. Загальним абревіатурою для дисперсійного аналізу є ANOVA (ANAlysis Of Variance) або дисперсійний аналіз.

Основним компонентом дисперсійного аналізу є аналіз значущості середньої різниці. Розмір вибірки менше одиниці (n-1), поділений на суму квадратів відхилень від середнього вибіркового значення, дає дисперсію вибірки для розміру вибірки n. Таким чином, сума квадратів відхилень, або SS (Sum of Squares), визначає дисперсію для фіксованого розміру вибірки n. Після обчислення дисперсії розробляється нульова гіпотеза щодо факторів, з якими

пов’язані зразки. Нульову гіпотезу можна підтвердити або спростувати після обчислення критерію Фішера та Р-значень.

Нижче описано, як зазвичай формуються гіпотези в дисперсійному аналізі:

– наступна нульова гіпотеза: «середні значення результуючої ознаки в усіх факторних (або градаційних) ситуаціях однакові»;

– «середні значення результуючої ознаки в різних умовах чинника є різними» — альтернативна гіпотеза.

Дисперсійний аналіз є рекомендованим методом під час вивчення впливу кількох якісних факторів на одну важливу кількісну змінну. Він працює на основі передумови, що деякі змінні є причинами, а деякі є наслідками. Вони також відомі як контрольні фактори, оскільки дослідник має можливість змінювати значення незалежних змінних і оцінювати, як це вплине на результати дослідження. Хоча методи оптимізації можна використовувати безпосередньо як незалежні змінні в цьому дослідженні, час обробки запиту може служити кількісною змінною, на яку впливають ці параметри.

Метою узагальненого дисперсійного аналізу є виявлення трьох різних дисперсій у загальній дисперсії.

– мінливість, викликана впливом кожної з досліджуваних незалежних змінних;

– мінливість, спричинена взаємодією між досліджуваними незалежними змінними;

– стохастична мінливість, викликана будь-якими непередбаченими обставинами.

Випадкова мінливість і мінливість, що виникає внаслідок факторів і їх взаємодії, корелюють. Ця кореляція вимірюється за допомогою критерію Фішера, який найчастіше позначається буквою F.

$F_{емп}$АБ = варіативність, що зумовлена взаємодією А і Б / Випадкова варіативність.

$F_{емп}$А = варіативність, що зумовлена дією змінної А / Випадкова варіативність.

$F_{емп}$Б = варіативність, що зумовлена дією змінної Б / Випадкова

варіативність.

Оскільки оцінки дисперсії є компонентом формули, що використовується для визначення F-критеріїв, цей метод є параметричним. Чим більше дисперсія характеристики пояснюється досліджуваними змінними або їх взаємодією, тим вищі значення емпіричного F-критерію.

Дисперсійний аналіз, на відміну від кореляційного аналізу, робить припущення, що деякі змінні, які також називають факторами або незалежними змінними, суттєво впливають на інші змінні, які також називають атрибутами результату або залежними змінними, і що ці фактори впливають на інші змінні. Хоча це припущення формує основу кількісних методів обчислення, воно потребує ретельного вивчення джерела та мети впливу.

Схематично існує кілька груп, на які можна розділити дисперсійний аналіз. Поділ ґрунтується на трьох аспектах: кількості змінних, на які зазвичай впливають фактори, кількості залучених факторів і зв’язку між вибірками значень. Коли досліджується вплив одного компонента, дисперсійний аналіз, який іноді називають однофакторним аналізом, має дві форми:

– дослідження різних або непов’язаних зразків. Наприклад, завдання виконує одна група учасників експерименту в тихій кімнаті, а інша група добровольців – у галасливій. (У цьому випадку нульовою гіпотезою буде: «Коефіцієнт шуму не впливає на час вирішення, оскільки середній час, необхідний для вирішення завдань цього типу як у тихому, так і в шумному середовищі, буде однаковим.»)

– експертиза порівнюваних зразків. Іншими словами, це відноситься до двох оцінок, проведених з використанням ідентичних елементів, але з різними параметрами. У тому ж прикладі шумові перешкоди призвели до того, що ідентичне завдання було виконано двічі, тоді як перше завдання було виконано без звуку. (Насправді такі тести слід проводити обережно, оскільки вони можуть пропустити вплив іншого елемента, відомого як «здатність до навчання», який дослідник наражається на небезпеку, приписуючи йому умовну варіацію — особливо шум.)

Однофакторний дисперсійний аналіз використовується, коли варіації

результату (залежна змінна) досліджуються у зв’язку зі змінами термінів або рівнів будь-якого фактора.

Кожен рівень фактора впливає на окрему вибірку.

Фактор повинен мати два спостереження на градацію та мінімум три градації.

Щоб почати обчислення, усі дані впорядковують у стовпці, що відповідає кожному фактору.

Наступним кроком є пошук і зведення сум стовпців або градацій.

Справжня процедура полягає в порівнянні сум квадратів усіх значень, отриманих під час експерименту.

Ми маємо справу з багатофакторним дисперсійним аналізом, коли досліджуємо одночасний вплив двох або більше компонентів; це можна класифікувати за методом вибірки. Коли багато змінних мають тенденцію піддаватися впливу багатьох причин, використовується багатофакторний аналіз [5].

2.5 Опис методів дослідження

Ефективність підходів до оптимізації бази даних можна оцінити за допомогою різноманітних параметрів, як було розглянуто в попередньому розділі. Більшість із цих факторів пов’язані з характеристиками конкретної програмної системи. Навіть якщо модель детермінована, вплив дослідження все одно можна збільшити за допомогою імовірнісної моделі. Це покращить застосовність дослідження в реальних умовах і підвищить його умовний ступінь універсальності.

Коли результати дослідження безпосередньо пов’язані з часом, вибір методології дослідження здійснюється на основі того, чи результати сумісні з моделлю. Як було сказано раніше, результати дослідження можуть бути дуже непередбачуваними, якщо умови, що оточують досліджуване явище, змінюються. В результаті повинні бути проведені експериментальні дослідження складових статистичних спостережень.

Статистичне спостереження, початковий етап статистичного дослідження, дозволяє методично збирати дані за допомогою попередньо встановленого процесу для використання в подальших статистичних дослідженнях.

Проте можна використати декілька статистичних спостережень для отримання результатів дослідження, застосовних до ширшого спектру обставин, ніж для окремого набору випадків. Це дослідження зосереджено на тому, скільки часу потрібно для обробки запитів до бази даних за допомогою засобів СУБД, тому важливо розрахувати вплив факторів оптимізації та визначити ступінь їхнього впливу, а також ймовірність отримання порівнянних результатів у реальних програмах. кінцевий результат визначається їх поєднанням. Однією зі стратегій, яка добре працює для отримання цього результату, є дослідження дисперсії.

3 ОПИС ПРОВЕДЕНИХ ДОСЛІДЖЕНЬ

3.1 Обрання предметної галузі та стеку технологій

Як було сказано раніше, без реального використання системи моделювання можна використовувати для імітації поведінки програмної системи або її частини. Ми повинні взяти до уваги методи оптимізації баз даних для програмних систем у світлі мети цієї статті.

Базуючись на статистичних даних, які дають повну картину використання різних типів баз даних, реляційні бази даних залишаються найбільш часто використовуваним типом баз даних на даний момент. Таблиці, що містять дані, зберігаються в реляційних базах даних і можуть бути точно пов’язані зі стовпцями таблиці за допомогою техніки введення ключа. Щоб прискорити вибір, користувач реляційної бази даних зазвичай може додавати індекси до таблиць зі структурою хеш-карти. Завдяки цьому пошук відповідного запису в таблиці тепер значно швидший. Найпопулярніші системи керування реляційними базами даних (RDBMS) — Oregon, MySql і PostgreSql — мають майже схожі структури, що робить їх порівнянними для визначення спільних аспектів, які слід включити в модель і, можливо, потребуватиме оптимізації.

Оскільки оптимізація бази даних є відносно новою ідеєю, важливо вибрати правильний тип бази даних, перш ніж заглиблюватися в цю тему далі. Сьогодні кількість онлайн-магазинів в Україні зростає через швидко зростаючий сектор онлайн-покупок. Через це зі збільшенням аудиторії зростає потреба в оптимізації, щоб підвищити залученість користувачів і підвищити рівень конкуренції роздрібного продавця. Як правило, найбільш багатим на функції розділом онлайн-бізнесу є його каталог, який містить низку фільтрів. Кожен продукт містить набір атрибутів, які дозволяють споживачеві магазину звузити каталог і знайти саме той товар, який йому потрібен. Отже, давайте використаємо базу даних типового інтернет-магазину як нашу модель дослідження. У цьому випадку основна увага приділятиметься таблицям із відомостями про продукт, типами фільтрів і даними фільтрів.

Оскільки більшість онлайн-бізнесу в Україні та світі побудовано на мові

програмування PHP та інших системах керування контентом, які базуються на мові програмування PHP, ми будемо використовувати MySQL, найпоширенішу систему керування базами даних (СУБД) у проектах PHP.

3.2 Формулювання критеріїв дослідження

Ми повинні визначити найпопулярніші типи запитів для показаних таблиць, щоб визначити фактори, що впливають на швидкість бази даних. Як було сказано раніше, причина, по якій запити типу SELECT будуть використовуватися, полягає в тому, що частина системи, спрямована на користувача, оптимізована. Одним із поширених методів покращення продуктивності запиту SELECT є створення індексів за атрибутами, які використовуються для пошуку рядків у таблиці або групі таблиць. Крім того, використання об’єднання для об’єднання таблиць у запиті або відсутність індексу для стовпця може серйозно погіршити продуктивність обробки запиту. Таким чином, спостерігаємо, що наявність індексів є одним із факторів, що впливають на швидкість обробки запитів СУБД.

Одним із найважливіших етапів створення бази даних є її нормалізація. Нормалізація — це процес зміни структури бази даних відповідно до специфікацій, які містяться в так званих нормальних формах. Основна мета нормалізації — зменшити надлишкові дані, зберігаючи їх цілісність. База даних, яка зазнала нормалізації, як правило, має змінені таблиці та структури зберігання даних. На перший погляд оптимальність і ступінь нормалізації здаються рука об руку; однак у певних ситуаціях високий ступінь нормалізації може значно перешкоджати здатності СУБД обробляти запити. Зазвичай це відбувається, коли таблиці об’єднуються за допомогою запитів SELECT через оператор JOIN. При об'єднанні таблиць за відношеннями «один-до-багатьох» або «багато-до-одного» (за допомогою оператора JOIN або вкладених запитів) рядки таблиці кількісно перемножуються, що призводить до відставання обробки запитів СУБД. Тому нормалізація значно впливає на швидкість обробки запитів до бази даних. У цьому відношенні недосконала нормалізація бази даних або її незалежного розділу — застосування другої нормальної форми — іноді забезпечує вищу

оптимальність. Таким чином, другим фактором, що впливає на продуктивність обробки запитів стороною СУБД, є рівень нормалізації бази даних.

Якщо припустити, що ці фактори є атомарними та між ними є статистична неточність, це спростить завдання.

3.3 Створення дослідницької програмної моделі

Як зазначалося раніше, технологічним стеком, обраним для дослідження, був звичайний стек PHP. Різноманітні веб-додатки, такі як онлайн-вітрини, інформаційні ресурси, системи обробки даних і агрегації, часто розробляються з використанням цього стеку. У цьому випадку використовувався добре відомий фреймворк MVC PHP Laravel. Для вибору СУБД використовувався MySQL.

Більшість програмних продуктів розробляється в послідовності розробки проекту незалежно від їх функціональних особливостей. Як правило, це кілька кроків, починаючи з розробки технічної специфікації та закінчуючи тестуванням готового продукту. У цьому випадку легко виділити чотири етапи створення проекту:

– розробка технічного завдання на проект;

– побудова серверної та бази даних архітектури проекту;

– використання попередньо вибраного технологічного стеку та технічної специфікації в реальних програмах;

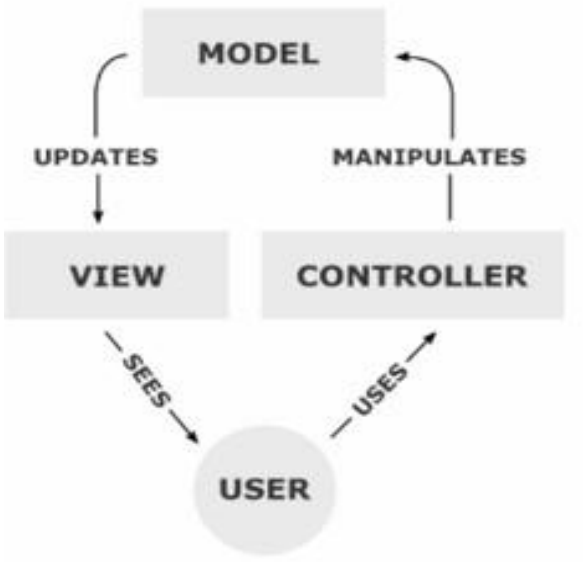
– підтверджуючи правильність реалізованої функції.

База даних та її конкретний вміст повинні розглядатися як основний предмет дослідження, оскільки основною метою цієї роботи є вивчення та порівняння різних методів вдосконалення програмних систем баз даних. Крім того, його потрібно модифікувати для реальних умов використання. У результаті було вирішено розробити веб-інтерфейс, наданий код сервера та систему програмного забезпечення бази даних у цьому випадку. Завдяки налаштуванню параметрів експерименту, бази даних і її вмісту їх можна використовувати для дослідження змінних, які впливають на швидкість обробки запитів СУБД. Оскільки жодна з функцій інтернет-магазину не може вплинути на хід або

результат дослідження, не потрібно повністю використовувати їх усі.

Оскільки фреймворк Laravel PHP є частиною технологічного стеку, архітектуру сервера проекту, яка базується на архітектурному шаблоні MVC, можна розділити на три архітектурні рівні. Модель безпосередньо відповідає за бізнес-логіку проекту та рівень доступу до даних, тоді як представлення відповідає за те, як системний інтерфейс виглядає та функціонує з користувачами. Рівень контролера відповідає за нагляд за системою та використання інших двох рівнів для формувати свої дії. Можливо, деякі розробники неправильно інтерпретують шари цього шаблону та вважають, що кожен контролер має стосуватися лише одного представлення чи об’єкта.

Оскільки ця помилка іноді може мати незворотні наслідки для дизайну проекту, слід чітко пояснити, що розділення зазвичай відбувається на рівні просторів імен або інших структурних одиниць проекту, а не на рівні окремого класу чи набору класів. Схематичне зображення цієї архітектурної конструкції можна знайти на малюнок 3.1.



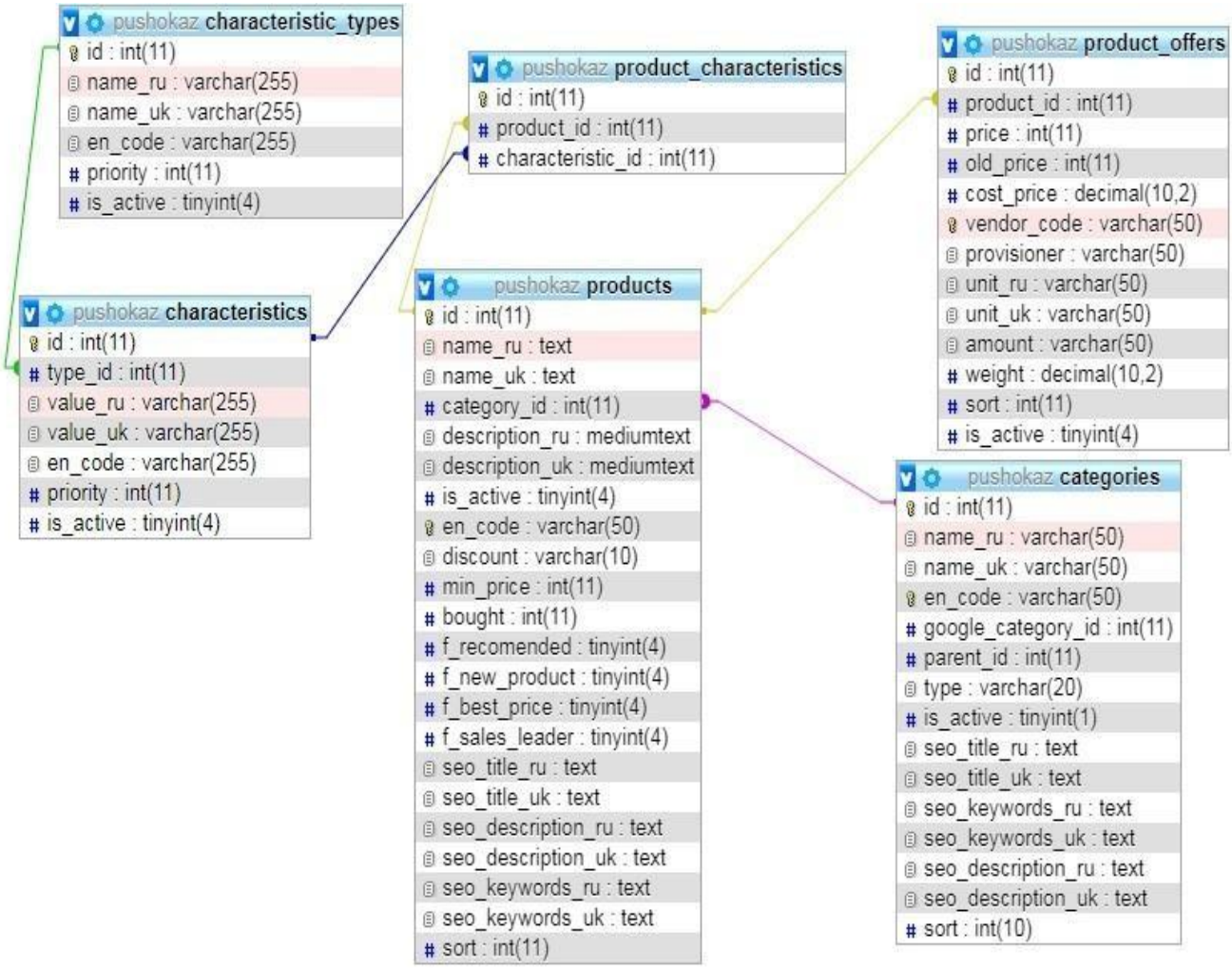
малюнок 3.1 – Схема MVC

Структура бази даних досліджуваної програмної системи схожа на структуру типового інтернет-магазину, як загального, так і спеціального призначення. Він містить таблиці, які містять дані про клієнтів, інформацію про

замовлення, інформацію про SEO, каталог магазину (товари, пропозиції, фільтри та групи фільтрів) і кілька допоміжних таблиць.

Схема сегмента бази даних, що містить дані каталогу, зображена на малюнок 3.2.

Щоб візуалізувати експеримент і покращити налаштування системи, було вирішено створити веб-інтерфейс, який повинен змінити конфігурацію експерименту, структуру бази даних і її безпосередній вміст.

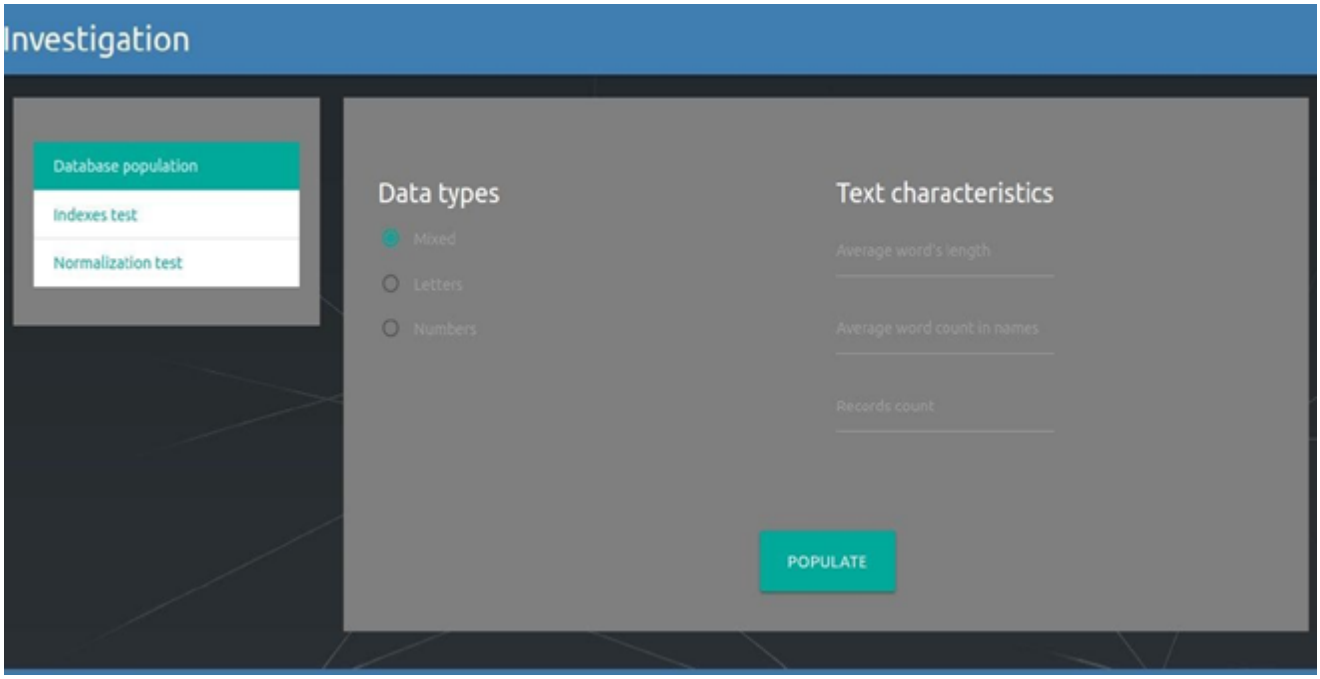


малюнок 3.2 – Фрагмент схеми БД, що використовувався у дослідженні

Моделюючи розподіл функцій інтерфейсу, було визначено три різні набори інструментів. Кожен із цих блоків контролює окрему галузь дослідження.

Особливо блок інструментів, який використовується для додавання тестів із відповідними характеристиками до бази даних. Серед них тип даних, середня довжина слова, середня кількість слів у заголовках і загальна кількість записів

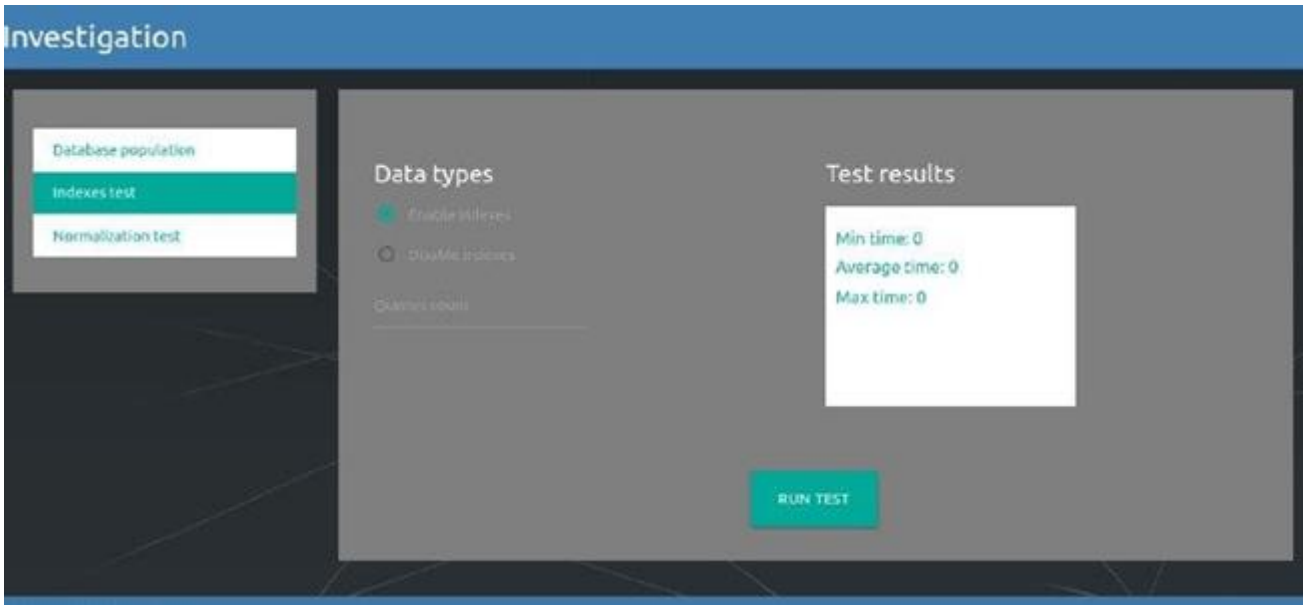
бази даних.



малюнок 3.2 – Блок налаштувань заповнення бази даних

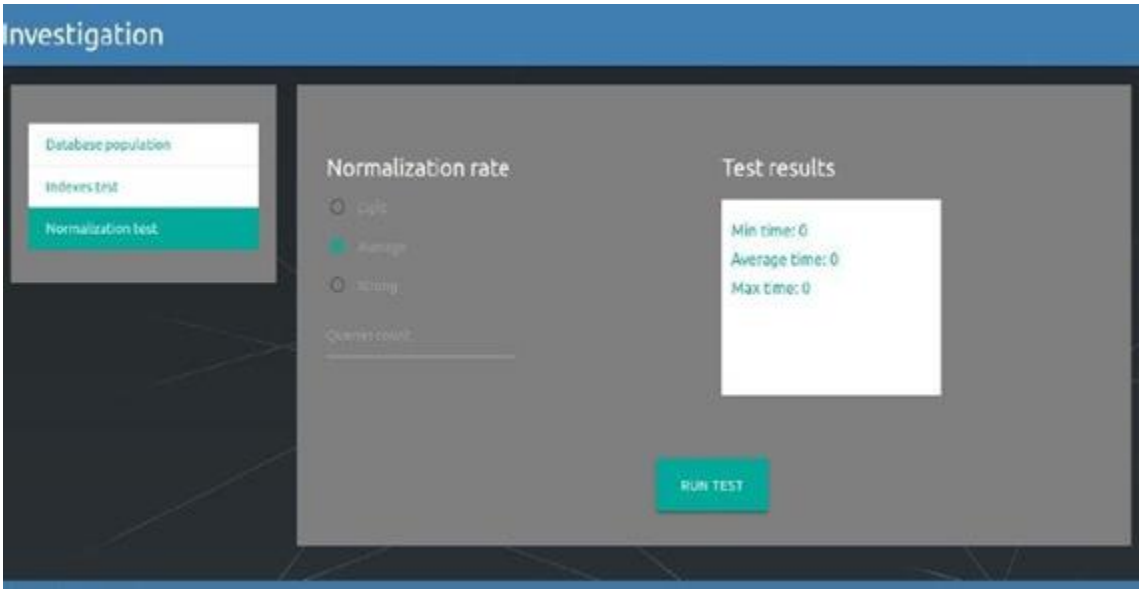
Для створення тестових даних для заповнення таблиць і набору атрибутів даних, таких як середня кількість слів у заголовку, формат даних, середня кількість літер у слові, тощо, було створено та введено в експлуатацію інший клас, призначений для реалізації заповнення експериментальної бази даних.

Блок налаштування індексації бази даних додає індекси до відповідних стовпців таблиць системної бази даних. Індекси додаються до доступних стовпців автоматично, коли таблиці з’єднуються за допомогою оператора JOIN. Цей блок зображено на малюнок 3.3.



малюнок 3.3 – Блок налаштування індексів у бази даних

Блок налаштування нормалізації бази даних показано на малюноку 3.4. Він відповідає за використання певної колекції таблиць, які повертаються до їхньої звичайної нормалізованої форми, натиснувши кнопку та налаштувавши положення перемикача. Іншим варіантом є кількість запитів, які потрібно виконати протягом пробного періоду. Блок індикації часу показує найнижчий, максимальний і середній арифметичний час виконання запиту до бази даних після експерименту.



малюнок 3.4 – Блок налаштування ступеню нормалізації бази даних

Для виконання цієї функції структуру таблиці бази даних було змінено шляхом виконання запитів на рівні моделі.

Використовуючи останні два інтерфейси, базу даних було налаштовано відповідно до плану експерименту. Щоб визначити результат експерименту та час обробки запиту на рівні СУБД, тестові значення були додані до бази даних і набір запитів був повторений.

3.4 Опис проведених експериментів

Дисперсійний аналіз був обраний як метод для отримання шаблону з даних дослідження, як обговорювалося в попередньому розділі. Враховуючи, що два параметри були вибрані для впливу на кінцевий час обробки SQL-запиту, у цій ситуації слід провести двофакторний дисперсійний аналіз. Необхідно вибрати фіксований набір незалежних змінних, і кожен експеримент повинен перевірити кожну можливу комбінацію значень факторів, щоб зробити двофакторний дисперсійний аналіз.

Два параметри, які будуть використані в цьому дослідженні, це ступінь нормалізації бази даних, точніше, її частина, яка буде використовуватися в експериментальних дослідженнях, і наявність індексів у стовпцях таблиці, які використовуються під час фільтрації даних або під час підключення. таблиці в запиті. тривалість виконання запиту. У цьому випадку для цього фактора буде використано три значення:

– слабка нормалізація (еквівалентно 1 NF);
– середня нормалізація (еквівалентно 2 NF);
– сильна нормалізація (еквівалентно 3 NF).

Перші три нормальні форми зазвичай використовуються для нормалізації бази даних, тому було обрано цей набір значень.

Слід зазначити, що експерименти починаються з внесення тестових значень до бази даних. Основними характеристиками тестового набору даних були:

– тип даних (текст, змішаний або числовий);
– типова кількість букв у слові;

– типова кількість слів в імені;

– кількість товару

Після проведення експерименту, щоб переконатися в якості тестових даних, було вирішено використовувати це співвідношення як основу. Зокрема, кількість характеристик має дорівнювати п’яти частинам від кількості товарів, а кількість категорій має приблизно дорівнювати двадцяти частинам від загальної кількості товарів. Було вирішено обмежити кількість продуктів до 1000, що приблизно дорівнює кількості продуктів у типовому онлайн-бізнесу. Для програмного заповнення бази даних було зроблено три кроки:

– заповнення товарних і категорійних таблиць;

– оновлення таблиць за допомогою характеристик (фільтрів);

– заповнення асоціативної таблиці з метою зв’язку ознак і сутностей товару в тип M-N.

Експериментальне дослідження почне розглядати кожну можливу комбінацію факторів впливу, зазначених у попередньому розділі, після заповнення бази даних. Було шість можливих комбінацій, оскільки фактори впливу мали два та три стани відповідно. Кожне випробування створювало колекцію значень для часу обробки запиту.

Щоб гарантувати точність експериментальних результатів і застосовність дисперсійного аналізу, у кожному експерименті використовувалися однакові 1000 запитів до бази даних. Включивши в код сервера функцію, яка дозволяє експортувати результати кожного експерименту у файл a.csv, ми змогли розрахувати середній час обробки запиту в секундах для використання в регресійному аналізі. На малюнок 3.5 зображено цю таблицю.

	A	B	C	D
1				
2				
3		A1	A2	A3
4	B1	0,000162	0,000152	0,0001785
5	B2	0,000165	0,000118	0,000147

малюнок 3.5 – Результат експериментів

Файл Excel містить огляд даних середнього часу для кожного випробування. Фактори експозиції (ступінь нормалізації та наявність індексів) позначимо як A та B відповідно, а окремі значення факторів позначимо як A1, A2, A3 та B1, B2.

4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ

4.1 Проведення дисперсійного аналізу

Ми будемо використовувати двофакторний дисперсійний аналіз на додаток до багатофакторного дисперсійного аналізу, як було описано раніше в розділах про аналіз зібраних експериментальних даних, оскільки в цьому випадку до уваги беруться два фактори впливу. Буде використано нульову гіпотезу, яка стверджує, що жоден із факторів впливу не впливає на рівень продуктивності обробки запитів СУБД.

Було вирішено провести дисперсійний аналіз даних за допомогою вбудованого модуля MS Excel. Щоб використовувати цей модуль, необхідно скористатися меню «Аналіз даних» на вкладці «Дані». Вікно меню «Аналіз даних»

Виберіть «Двофакторний дисперсійний аналіз без повторення» в меню інструментів аналізу даних, а потім виберіть інтервал вхідних значень. Крім того, ми встановлюємо набір Alpha (значний рівень), який, по суті, є показником точності експериментальних даних. З дуже скромним рівнем помилки 28% як найбільшим припущенням помилки ми встановили значення Alpha 0,28.

Рівень похибки обраної моделі 28% вказує на те, що вона достатньо точна для пояснення цієї системи, яка все ще потребує оптимізації. Параметр аналізу дисперсії.

Після дисперсійного аналізу MS Excel створює таблиці з визначеними значеннями дисперсії разом з іншими показниками. Таблиця 4.1 пропонує скорочений перелік цих характеристик.

Таблиця 4.1 – Результат дисперсійного аналізу

Результати	Рахунок	Сума	Середнє	Дисперсія
Рядок 1	3	0,0004923	0,000164167	1,79083E-10
Рядок 2	3	0,000421	0,000143333	5,62333E-10
Стовпчик 1	2	0,000325	0,0001635	4,61E-12
Стовпчик 2	2	0,00025	0,000135	5,78E-10
Стовпчик 3	2	0,0003255	0,00016275	4,96226E-10

Ця таблиця демонструє, як MS Excel обчислює індекс дисперсії для кожного рядка та стовпця таблиці вхідних даних після незалежного визначення дисперсії кожного рядка та стовпця.

У таблиці показано середнє значення, дисперсію та суму вхідних даних для кожного рядка та стовпця.

У остаточному аналізі дисперсії немає таблиці. Два найважливіші дисперсійні аналізи, P-value і критерії Фішера, також обчислюються за допомогою MS Excel. У таблиці додатково показано середньоквадратичне значення. P-значення, ступені свободи та результати критерію Фішера в рядках і стовпцях відображені в таблиці 4.2, яка також була створена за допомогою обчислень Microsoft Excel.

Таблиця 4.2 – Результат дисперсійного аналізу

Джерело варіації	SS	df	F	P-Значення	F критичне
Рядки(B)	6,51042E-10	1	3,045215356	0,223092793	2,033071
Стовпці(A)	1,05523E-09	2	2,467939973	0,288355623	2,44825
Похибка	4,27583E-10	2			
Разом	2,13386E-09	5			

У цій таблиці наведено огляд P-value і розрахунків за критерієм Фішера.

4.2 Аналіз результатів експерименту

Відповідно до таблиці 4.2 значення F нижче критичного порогу. У цьому випадку це свідчить про те, що фактори впливу мають суттєвий вплив на продуктивність обробки запитів до бази даних і що нульова гіпотеза щодо них розвінчана.

Відповідно до результатів експерименту, якщо порядок даних такий самий, як і експериментальні дані, і використовується структура бази даних, подібна до експериментальної бази даних, ступінь нормалізації матиме більший вплив на швидкість обробки запиту, ніж наявність індексів. P-value показує ймовірність того, що змінна, яка використовується для обчислення P-value, впливає на результат експерименту. Інакше кажучи, в даному випадку на результат експерименту впливатиме рівень нормалізації бази даних з імовірністю близько 0,28 і наявність індексів у необхідних стовпцях таблиць бази даних з імовірністю 0,22.

4.3 Практичне застосування результатів дослідження

Як зазначалося раніше, дуже ймовірно, що будь-який обраний критерій суттєво вплине на швидкість обробки SQL-запитів; однак справжня корисність цього дослідження полягає в іншому. У таблиці 4.3 наведено статистичний аналіз комбінацій факторів.

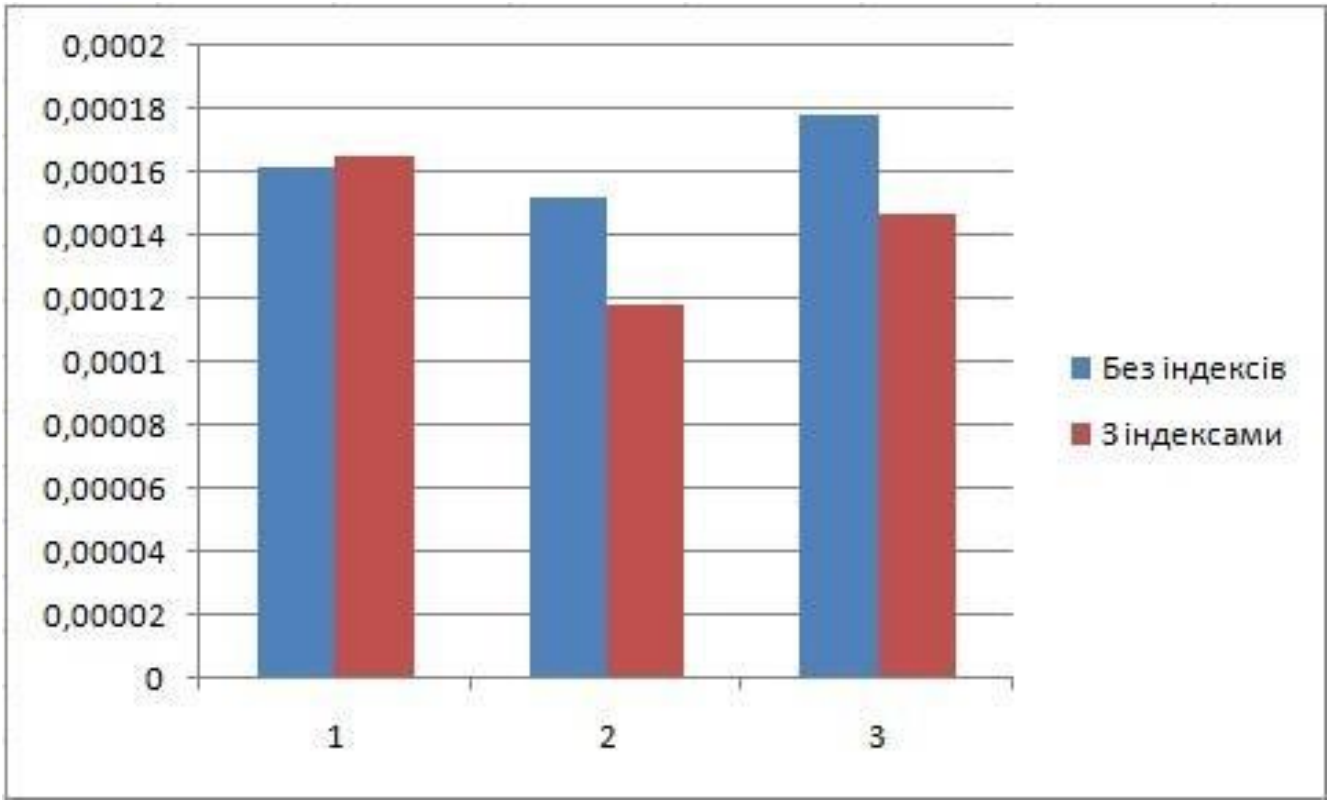
Щоб забезпечити візуальне зображення результатів, ми використовуємо MS Excel для створення гістограми даних, представляючи фактори A і B горизонтально та інколи вертикально. Гістограма показана у таблиці 4.3.

Таблиця 4.3 – Результати експериментів

Фактори	A1	A2	A3
B1	0,000162	0,000152	0,0001783
B2	0,000163	0,000118	0,000147

На графіку видно, що коли були індекси та певний рівень денормалізації, або при 2NF, було видно найкращу індикацію часу. Це свідчить про те, що використання цієї колекції змінних оптимізації з налаштуваннями, схожими на експериментальні, було б найкращим підходом для прискорення обробки запитів SQL на рівні СУБД. Це відкриття може бути корисним у розробці систем подібної архітектури.

Результати 1NF справді яскраві. Було виявлено, що наявність індексів збільшує час обробки запиту порівняно з їх відсутністю. Причиною такого результату можуть бути особливості архітектури індексу.



малюнок 4.3 – Гістограма з результатами експериментів

Таблиця значень, яка краще працює на складній колекції різних значень, називається індексом. У деяких випадках обробка запиту значно сповільнюється індексом, який часто використовується для прискорення обробки запиту. Результатом є результат значно повільнішого зчитування індексу через велику кількість дублікатів, що також заважає індексу знайти унікальне значення. Після

прочитання індексу СУБД починає прямий пошук значення в таблиці, якщо вона не може знайти унікальне значення в індексі.

Цей результат свідчить про те, що не всі взаємодії з базою даних сповільнюються через використання індексу та що база даних може бути надмірно нормалізована.

ВИСНОВКИ

Дослідження охоплювало фундаментальні прийоми та етапи математичного моделювання системи реляційної бази даних. Враховуючи обсяг статистичних даних, було визначено, що найкращою математичною моделлю для цієї конкретної ситуації була статистична модель із припущенням часу результату.

Після ретельного вивчення впливу індексу та рівня нормалізації бази даних на продуктивність обробки запитів СУБД було проведено двофакторний дисперсійний аналіз для підтвердження результатів статистичного дослідження. Результати продемонстрували, що два критерії були належним чином зважені, щоб суттєво вплинути на швидкість обробки SQL-запитів.

Також було досліджено вплив вищезазначених факторів на значні колекції змішаних даних, які регулярно використовуються.

Швидшої взаємодії з базою даних можна досягти, застосувавши результати дослідження до реальних додатків із використанням стандартно структурованої бази даних від онлайн-магазину.

Крім того, результати дослідження можна використовувати як дорожню карту для розробки нових реляційних баз даних із подібним набором функцій.

Згодом завдання дослідження потребуватиме більшої складності через включення інших факторів, які впливають на швидкість обробки запитів SQL. Також потрібно буде перевірити, чи змінюється рівень впливу факторів залежно від набору тестових даних, який використовується.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Silvia Botros, Jeremy Tinley. High Performance MySQL: Proven Strategies for Operating at Scale / Publisher : O'Reilly Media, 2021. 386 p.

2. Основи наукових досліджень : навч. посібник / Невлюдов І. Ш., Олександров Ю. М., Андрусевич А. О., Чала О.О. Кривий Ріг: Криворізький коледж НАУ, 2019. 396 с. URL: https://openarchive.nure.ua/server/api/core/bitstreams/d4ed456f-4126-4ee0-a20b-91dec7528f0e/content (дата звернення: 09.05.2024)

3. A. A. Afifi, S. P. Azen. Statistical Analysis: A Computer Oriented Approach / Publisher : Academic Press, 2015. 384 p.

4. MySQL Documents. URL: https://dev.mysql.com/doc/ (date of access : 10.05.2024)

5. Joseph B. Miller. Internet Technologies and Information Services / Publisher : Libraries Unlimited, 2014. 520 p.

6. Martin Kleppmann. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems / Publisher : O'Reilly Mediam, 2017. 611 p.

7. Jan L. Harrington. Relational Database Design and Implementation: Clearly Explained / Publisher : Morgan Kaufmann, 2016. 712 p.

8. Eric Vanier, Birju Shah, Tejaswi Malepati. Advanced MySQL 8: Discover the full potential of MySQL and ensure high performance of your database / Publisher: Packt Publishing, 2019. 286 p.

9. Tadeusz Morzy, Theo Härder, Robert Wrembel. Advances in Databases and Information Systems / Publisher : Springer, 2013. 328 p.

10. WANG, Xing,. Database optimization concepts in fast response environments. U.S. Patent Application No 15/592,709, 2017.

11. A Real-Time Database Survey: The Architecture of Meteor, RethinkDB, Parse &

Firebase. URL: https://medium.baqend.com/real-time-databases-explained why-meteor-rethinkdb-parse-and-firebase-dont-scale-822ff87d2f87 (date of access : 10.05.2024)

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ 42 НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

1. Каук В., Гребенюк В., Пуголовок К., Водяницький Д. Виклики, які надають нові можливості // Екстренне дистанційне навчання в Україні : монографія. 2020. С. 223-232.

3. Афанасьєва І. В., Перов О. С. Фреймворк для рендерінгу 3d сцен на платформах, що підтримують Metal API // Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління : тез. доп. дванадцатої міжнародної науково-технічної конференції, 27–28 квітня 2022 р. Т. 2. Баку – Харків – Жиліна, 2022. С. 117.

9. Bielievtsov, I. Ruban, K. Smelyakov, D. Sumtsov Network technology for transmission of visual information // Selected Papers of the XVIII International Scientific and Practical Conference on Information Technologies and Security (ITS 2018). CEUR Workshop Processing. Kyiv, Ukraine, November 27, 2018. Pp. 160- 175.

11. Leshchynskyi Volodymyr. Principles of explanation in e-commerce system based on sales dynamics. Computer and information systems and technologies. URL: http://litopys.chdu.edu.ua/index.php/wissn100/article/view/201452 (date of access : 10.05.2024)