

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається  
Т.в.о завідувач кафедри  
\_\_\_\_\_ Сафонова С.О.  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**МАГІСТЕРСЬКА РОБОТА**

НА ТЕМУ:

«ДОСЛІДЖЕННЯ МЕТОДІВ ОПТИМІЗАЦІЇ ЗАПИТІВ ДО БАЗИ ДАНИХ»

---

---

---

Освітньо-кваліфікаційний рівень “Магістр”  
Спеціальність 123 - “Комп’ютерна інженерія”

Науковий керівник роботи:

\_\_\_\_\_ (підпис)

Нестеров М.В.

\_\_\_\_\_ (ініціали, прізвище)

Консультант з охорони праці:

\_\_\_\_\_ (підпис)

Критська Я.О.

\_\_\_\_\_ (ініціали, прізвище)

Студент:

\_\_\_\_\_ (підпис)

Бакітько Д.Е.

\_\_\_\_\_ (ініціали, прізвище)

Група:

\_\_\_\_\_ КІ-18дм

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки  
Кафедра Комп'ютерних наук та інженерії  
Освітньо-кваліфікаційний рівень магістр  
Напрямок підготовки \_\_\_\_\_  
(шифр і назва)  
Спеціальність 123 «Комп'ютерна інженерія»  
(шифр і назва)

**ЗАТВЕРДЖУЮ:**

Т.в.о завідувач  
кафедри \_\_\_\_\_  
Сафонова С.О.  
« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**З А В Д А Н Н Я  
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Бакітько Денису Едуардовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів оптимізації запитів до бази даних»

керівник проекту (роботи) к.т.н., ст. викл. Нестеров М.В.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від " " \_\_\_\_\_ 2019 р. № \_\_\_\_\_

2. Строк подання студентом роботи \_\_\_\_\_

3. Вихідні дані до роботи матеріали з переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз методів і засобів оптимізації запитів до бази даних.  
постановка задачі досліджень, аналіз проблем оптимізації запитів, розробка  
методики фіксування повільних запитів, охорона праці та безпека в  
надзвичайних ситуаціях

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
Електронна презентація

## 6. Консультанти розділів проекту (роботи)

| Розділ  | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|---|---|----------------|------------------|
|   |   | завдання видав | завдання прийняв |
| Охорона праці та безпека в надзвичайних ситуаціях | ст. викл. Критська Я.О.                   |                |                  |
|   |   |                |                  |
|   |   |                |                  |
|   |   |                |                  |

7. Дата видачі завдання \_\_\_\_\_

Керівник \_\_\_\_\_  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проекту (роботи)  | Строк виконання етапів проекту ( роботи ) | Примітка |
|-------|---|---|----------|
| 1     | Аналіз програмних засобів для моніторингу станів бази даних                           | 03.09 – 08.09                             |          |
| 2     | Порівняння програмних засобів   | 08.09 – 11.09                             |          |
| 3     | Аналіз та вибір метрик для фіксування показників бази даних                           | 18.09 – 25.09                             |          |
| 4     | Вибір математичного методу для розрахунку ймовірностей метрик                         | 02.10 – 14.10                             |          |
| 5     | Розрахунок порогових значень та ймовірностей методів оптимізації                      | 14.10 – 16.10                             |          |
| 6     | Розрахунок теоретичних показників використання системних ресурсів методам оптимізації | 16.10 – 07.11                             |          |
| 7     | Вибір методу оптимізації запиту   | 08.11 – 14.11                             |          |
| 8     | Розробка методики для фіксування та аналізу повільних запитів                         | 17.11 – 19.12                             |          |
| 9     | Оформлення  | 22.12 – 11.01                             |          |
|       |   |   |          |
|       |   |   |          |
|       |   |   |          |
|       |   |   |          |

Студент \_\_\_\_\_  
( підпис )

Науковий керівник \_\_\_\_\_  
( підпис )

Бакітько Д.Е  
(прізвище та ініціали)

Нестеров М.В  
(прізвище та ініціали)

## АНОТАЦІЯ

**Бакітько Д.Е. Дослідження методів оптимізації запитів до бази даних.**

Розглянуті програмні засоби для аналізу стану бази даних. Виявлено, що розглянуті засоби мають недолік несвоєчасного реагування на пікові навантаження. Було прийнято рішення розробити методику для виявлення повільних запитів та виявити найбільш продуктивний метод оптимізації запитів. З використанням математичного методу розраховані порогові значення для метрик та їх теоретичні показники продуктивності. Обрано математичний метод для оптимізації SQL-запиту згідно з розрахованих ймовірностей та використаних системних ресурсів.

**Ключові слова:** база даних, повільний запит, індексування, секціонування, оптимізація, системні ресурси, метрики.

## АННОТАЦИЯ

**Бакитко Д.Е. Исследование методов оптимизации запросов к базе данных.**

Рассмотрены программные средства для анализа состояния базы данных. Выведено, что рассмотренные средства имеют недостаток несвоевременного реагирования на пиковые нагрузки. Было принято решение разработать методику для выявления медленных запросов и выявить наиболее продуктивный метод оптимизации запросов. Выбран математический метод для оптимизации SQL-запросов согласно с рассчитанными вероятностями и использованием системных ресурсов.

**Ключевые слова:** база данных, медленный запрос, индексирования, секционирования, оптимизация, системные ресурсы, метрики.

## ABSTRACT

**Bakitko D.E. Research methods for optimizing database queries.**

Software tools for analyzing the state of the database are considered. It was revealed that the considered means have a lack of timely response to peak loads. It was decided to develop a methodology for identifying slow queries and to identify the most productive method for optimizing queries. Using the mathematical method, threshold values for metrics and their theoretical performance indicators are calculated. According to the probabilities and analysis of the use of system resources, a method was chosen to optimize the query.

**Key words:** database, slow query, indexing, partitioning, optimization, system resources, metrics

## ЗМІСТ

|   |           |
|---|-----------|
| <b>ВСТУП.....</b>   | <b>6</b>  |
| <b>РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ І ЗАСОБІВ ОПТИМІЗАЦІЇ ЗАПИТІВ ДО БАЗИ ДАНИХ. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕНЬ.....</b> | <b>8</b>  |
| 1.1 Аналіз вимог.....   | 8         |
| 1.2 Аналіз програмних та інструментальних засобів.....  | 8         |
| 1.2.1 SQL Sentry.....   | 8         |
| 1.2.2 System Center Operations Manager.....   | 11        |
| 1.2.3 Idera DB Optimizer Product.....   | 13        |
| 1.2.4 SolarWinds.....   | 14        |
| 1.2.5 Quest Software.....   | 15        |
| 1.2.6 Oracle Enterprise Manager.....  | 16        |
| 1.3 Постановка наукової задачі та обґрунтування методики досліджень.....                                      | 19        |
| Висновки до першого розділу.....  | 20        |
| Література до першого розділу.....  | 21        |
| <b>РОЗДІЛ 2 АНАЛІЗ ПРОБЛЕМ ОПТИМІЗАЦІЇ ЗАПИТІВ.....</b>   | <b>23</b> |
| 2.1 Метрики для підвищення продуктивності.....  | 23        |
| 2.2 Виявлення трудомістких SQL запитів.....   | 25        |
| 2.2.1 SHOW PROCESSLIST.....   | 26        |
| 2.2.2 Query Log.....  | 27        |
| 2.3 Виявлення повільного запиту та вибір математичного методу.....  | 28        |
| 2.3.1 Розрахунок показників метрик.....   | 30        |
| 2.3.2 Розрахунок оптимізації запитів.....   | 34        |
| 2.3.3 Планування оптимізації запиту.....  | 38        |
| Висновки до другого розділу.....  | 42        |
| Література до другого розділу.....  | 43        |
| <b>РОЗДІЛ 3. РОЗРОБКА МЕТОДИКИ ФІКСУВАННЯ ПОВІЛЬНИХ ЗАПИТІВ.....</b>  | <b>46</b> |
| 3.1 Проектування веб-інтерфейсу методики.....   | 46        |
| 3.2 Фіксування повільних запитів розробленою методикою.....   | 46        |
| 3.3 Аналіз не оптимізованого стану запитів.....   | 49        |
| 3.4 Аналіз запитів з методом індексування.....  | 50        |
| 3.5 Аналіз запитів з методом секціонування.....   | 52        |
| 3.6 Використання системних ресурсів.....  | 53        |

|  |           |
|--|-----------|
| <b>Висновки до третього розділу.....</b>   | <b>56</b> |
| <b>Література до третього розділу.....</b>   | <b>57</b> |
| <b>РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....</b>                                    | <b>58</b> |
| <b>4.1 Загальні питання з охорони праці.....</b>   | <b>58</b> |
| <b>4.1.1 Правові та організаційні основи охорони праці.....</b>  | <b>59</b> |
| <b>4.1.2 Організаційно-технічні заходи з безпеки праці.....</b>  | <b>60</b> |
| <b>4.2 Аналіз стану умов праці.....</b>  | <b>61</b> |
| <b>4.2.1 Вимоги до приміщень.....</b>  | <b>61</b> |
| <b>4.2.2 Вимоги до організації місця праці.....</b>  | <b>61</b> |
| <b>4.2.3 Навантаження та напруженість процесу праці.....</b>   | <b>62</b> |
| <b>4.3 Виробнича санітарія.....</b>  | <b>63</b> |
| <b>4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації)</b><br><b>виробу.....</b> | <b>63</b> |
| <b>4.3.2 Пожежна безпека.....</b>  | <b>63</b> |
| <b>4.4 Гігієнічні вимоги до параметрів виробничого середовища. Освітлення.....</b>                         | <b>64</b> |
| <b>4.5 Вентилювання.....</b>   | <b>67</b> |
| <b>4.6 Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).....</b>                      | <b>67</b> |
| <b>4.7 Екологія.....</b>   | <b>68</b> |
| <b>Висновки до розділу 4.....</b>  | <b>69</b> |
| <b>Література до четвертого розділу.....</b>   | <b>70</b> |
| <b>ВИСНОВКИ.....</b>   | <b>72</b> |
| <b>Додаток А – Схема алгоритму.....</b>  | <b>75</b> |
| <b>Додаток Б – Лістинг методики.....</b>   | <b>77</b> |
| <b>Додаток В – Презентація.....</b>  | <b>88</b> |

## ВСТУП

Час виконання запиту визначає продуктивність бази даних. Все більше додатків використовують базу даних. Дані ростуть, що призводить до уповільнення обробки. Довге виконання запиту несе навантаження на базу даних. З огляду на те, що додаток може використовуватися великою кількістю людей, додаток може працювати нестабільно і обробляти запити занадто довго. Повільний час відгуку SQL-запиту збільшує споживання системних ресурсів. Повільний SQL-запит споживає більше системних ресурсів, ніж виділено сконфігурованому примірнику.

Вирішуються завдання пошуку оптимального методу, який дозволить швидко виконати запит до бази даних, при цьому споживаючи мінімум системних ресурсів. Також правильна побудова структури бази даних є одною з основних стадій оптимізації запитів до бази даних. В даному дипломному проекті будуть розглянуті методи оптимізації запитів, завдяки яким значно покращиться швидкодія бази даних.

William Charles Eidson та Jesse Collins в своєму патенті «Methods and systems for joining indexes for query optimization in a multi-tenant database» звертали увагу на те, що швидкість обробки запиту до бази даних можна пришвидшити за допомогою індексування. [1]

John F. Hornibrook, Dieu Quang La, Calisto Paul Zuzarte в своєму патенті «Optimization of aggregate queries in database management systems using an early out join when processing min and max functions» звертають увагу на те, що прискорити запити до бази даних можна шляхом об'єднання запитів [2].

Xing Chen, Qing Yun Hao, Yi Jin, Wan Chuan Zhang в своєму патенті «Database query in a share-nothing database architecture» [3] звертають увагу на те, що покращити оптимізацію бази даних можна за допомогою запиту та його плану. На основі плану запиту будується оптимізаційний план доступу до ресурсів. План зберігається в зовнішньому ресурсі даних та містить в собі очікування розподілу зовнішніх даних. Спосіб також включає в себе отримання інформації про розподіл даних, пов'язаної з очікуваним розподілом зовнішніх даних, на основі оптимізованого плану доступу, передачу інформації про розподіл даних у зовнішнє джерело даних, так що зовнішнє джерело даних розділяє зовнішні дані відповідно до розподілу даних та повертає розділені зовнішні дані паралельно, і виконує пов'язану із запитом обробку розділених зовнішніх даних відповідно до оптимізованим планом доступу.

Тому обґрунтованою є тема магістерської роботи, у якій вирішується **науково-прикладне завдання** розроблення моделей і методу інформаційної технології оптимізації запитів до бази даних.

Об'єкт дослідження - процеси забезпечення оптимізації запитів до бази даних.

Предмет дослідження - моделі та метод інформаційної технології оцінки і забезпечення

оптимізації до бази даних з урахуванням структури бази.

**Мета і завдання дослідження.** Метою дослідження є підвищення швидкості виконання запиту до бази даних з урахуванням обраного методу.

Для досягнення мети дослідження необхідно вирішити такі **завдання**:

- аналіз програмних засобів для оптимізації бази даних;
- огляд математичних моделей.
- аналіз методів оптимізації запиту;
- впровадження методів та ймовірностей до бази даних.

**Методи дослідження.** Проведені в роботі дослідження основані на методах індексування, секціонування, які використовувались при розробленні (тестуванні) швидкості обробки запитів до бази даних.

**Особистий внесок здобувача** полягає у розробленні нових моделей, методів та інструментальних засобів, що дозволяють вирішити поставлені задачі. Усі основні результати отримані автором особисто. У роботах, опублікованих у співавторстві, автору належать: форум «IT-Idea 2018» (тези на тему «ОНЛАЙН-СИСТЕМА МІЖФАКУЛЬТЕТНОГО ЗВ'ЯЗКУ») [4], конференція «Майбутній науковець» 2018 (тези «АНАЛІЗ ПРИДАТНОСТІ ІСНУЮЧИХ ОНЛАЙН-СИСТЕМ ЗВ'ЯЗКУ ДЛЯ ВИКОРИСТАННЯ У МЕЖАХ УНІВЕРСИТЕТУ») [5], конференція «TACSIT-2019» (стаття «Query Optimization in Database Systems») [6, 7].

**Апробація матеріалів магістерської.** Основні положення, ідеї, висновки магістерської роботи доповідалися та обговорювалися на конференції TACSIT-2019.

**Зв'язок з науковими програмами, планами, темами.** Магістерська робота виконана у Східноукраїнському національному університеті ім. В. Даля у відповідності з державними програмами і планами НДР: Дослідження методів аналізу даних в медицині (Реєстрац. № 0116U008700)

**Практичне значення отриманих результатів** полягає в тому, що основні наукові положення дисертації реалізовані у виді розрахункових моделей та програмних засобів, які утворюють прикладну інформаційну технологію оптимізації запитів до бази даних.

**Публікації.** За темою магістерської роботи з викладенням її основних результатів опубліковано 3 наукові праці, серед яких 1 стаття у науковому фаховому виданні України; 2 тези доповідей міжнародних конференцій.

**Структура та обсяг магістерської.** Магістерська робота складається із вступу, чотирьох розділів, висновків, списку використаних джерел і додатків. Загальний обсяг дисертації складає 97 сторінок, з яких основний текст на 64 сторінках, список використаних джерел із 59 найменувань на 6 сторінках. Робота містить 20 таблиць та 46 рисунків.



# РОЗДІЛ 1

## АНАЛІЗ МЕТОДІВ І ЗАСОБІВ ОПТИМІЗАЦІЇ ЗАПИТІВ ДО БАЗИ ДАНИХ. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕНЬ

### 1.1 Аналіз вимог

Об'єктом дослідження є оптимізація запитів до бази даних.

Час виконання запиту визначає продуктивність бази даних. Все більше додатків використовують базу даних. Дані ростуть, що призводить до уповільнення обробки. Довгий виконання запиту несе навантаження на базу даних. З огляду на те, що додаток може використовуватися великою кількістю людей, додаток може працювати нестабільно і обробляти запити занадто довго. Повільний час відгуку SQL-запиту збільшує споживання системних ресурсів через поганий SQL-запит, поганого проектування бази даних. Поганий SQL-запит споживає більше системних ресурсів, ніж виділено сконфігурованого примірника.

Вирішуються завдання пошуку оптимального методу, який дозволить швидко виконати запит до бази даних, при цьому споживаючи мінімум системних ресурсів. Також правильна побудова структури бази даних є одною з основних стадій оптимізації запитів до бази даних. В даному дипломному проекті будуть розглянуті методи оптимізації запитів, завдяки яким значно покращиться швидкодія бази даних.

### 1.2 Аналіз програмних та інструментальних засобів

#### 1.2.1 SQL Sentry

Для моніторингу SentryOne був розроблений продукт SQL Sentry експертами SQL Server. Мета створення продукту – заощадження часу та покращення продуктивності баз даних.

SQL Sentry має панель, завдяки якій користувач інтуїтивно швидко зможе знайти інструменти, які призначені для огляду середовища та графічного відображення проблем та станів бази даних.



Рисунок 1.1 - Інтерфейс програмного засобу SQL Sentry [8]

Для адміністраторів баз даних дуже важливо мати можливість переглядати дані та ефективність використання середовища. В SQL Sentry є інформування про середовище, яке знаходиться на спеціальній панелі. Панель складається з [8]:

- а) Показань даних для всіх екземплярів SQL Server, які необхідно відстежити.
- б) Масштабувань для підтримки моніторингу сотень екземплярів сервера.
- в) Можливості керувати своїм оточенням на правильному рівні деталізації.
- г) Потужності та глибини приладної панелі аналізу ефективності SentryOne.

Інформаційна панель відображає показники продуктивності баз даних. Визначені такі показники, як навантаження та життєвий цикл екземплярів SQL. Панель містить показники операційної системи користувача (або віртуальну машину) та показники моніторингу серверу середовища. Панель складається з двох режимів:

- Режим «Демонстрація» («Demonstration»). Відстеження активності користувачів середовища за певний період часу.
- Режим «Історія» («History»). Ефективність роботи операційної системи та екземпляру SQL Server за певний період.

З використанням режиму «Історія» є можливість будувати лінію, яка графічно відображає дані про роботу середовища за певний проміжок часу та за вибраними критеріями продуктивності. Базова лінія також інформує про попереджувальні індикатори для різних станів середовища в SentryOne.

Панель розділена на розділи [8]:

- «Network». Мережевий трафік та інформація про контроль серверу.
- «CPU». Інформація про використання процесора сервера, навантаження на кожне ядро, загальна кількість використаних системних ресурсів.
- «System Memory». Об'єм використаної пам'яті процесами на сервері, а також помилки контексту та даних файлів контексту.
- «Disk I/O». Відображення проблем з читанням/запису кожного фізичного диску серверу.
- «SQL Server Activity». Стан кожного екземпляру серверу: блокування процесів, середнє значення транзакції та ін.
- «Wait». Класи та категорії очікувань які буди виконані в певних проміжок часу. Значення вимірюється в мс.
- «Database Memory». Використання виділеної пам'яті для кожного екземпляру з урахування розміру кеш-пам'яті буферу, об'єму кеш пам'яті та затримку зчитування/запису запитів (або відкладених запитів) до бази даних.

Для швидкого знаходження проблем під час виконання SQL запитів в SQL Sentry існує інструмент Top SQL.

Top SQL графічно відображає запити до бази даних. За допомогою відображення можна виявити проблеми виконання запиту та на ранніх стадіях вирішити проблеми оптимізації. Сортувальні сітки інструменту відображають діаграми виконання запитів, аналіз індексування. Статистика допоможе швидко виявити проблему.

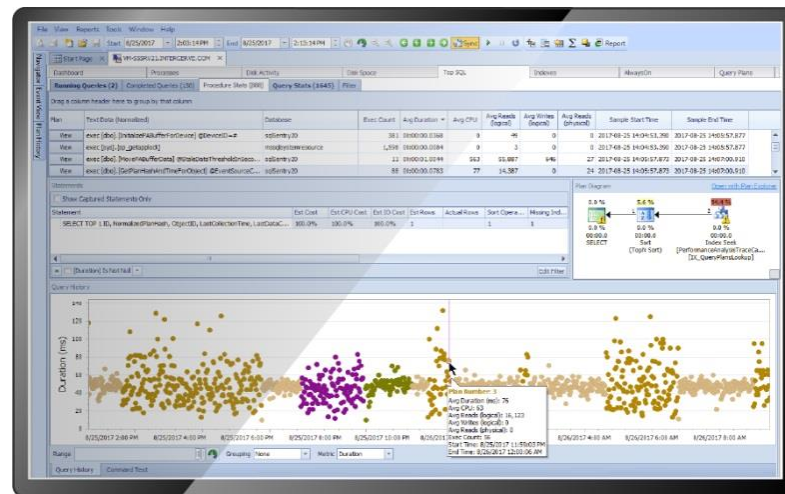


Рисунок 1.2 - Інтерфейс функції TOP SQL [8]

Top SQL є популярною функцією в SQL Sentry. Моніторинг за виконанням запитів має великий вплив на продуктивність бази даних. Можливості функції [8]:

- Перелічення пакетів T-SQL.
- Збереження процедур та операторів SQL.
- Перевірка перевищення виділених ресурсів для процедур (певний відсоток ЦП, час читання/запису).
- Аналіз проблемних запитів за певний проміжок часу (щохвилини, щогодини, щодня, щотижня та ін..)
- Перегляд тривалості запиту та використання кожним запитом системних ресурсів.
- Перегляд кількості виконаних операцій зчитування/запису.
- Швидкий перегляд плану запитів, що виконуються за запитом чи процедурою.

План можна відкривати, аналізувати та змінювати.

SQL Sentry має систему сповіщень (Advisory Conditions), яка інформує користувача лише о важливих змінах в середовищі та в випадках непродуктивного виконання SQL запиту. Advisory Conditions відображає актуальну інформацію в актуальний час. Завдяки такій моделі сповіщень у користувача не буде стомлення від великої кількості сповіщень незначних змін в середовищі. Система сповіщень має можливість запустити процедуру для виправлення ситуації та в питаннях оптимізації середовища. Advisory Conditions реагує на будь-які умови бази даних операційної системи користувачів, віртуальної машини (VM) або SQL Server. Advisory Conditions дозволяє створити нову умову, відредагувати існуючу умову та переглянути журнал

подій стану.

SQL Sentry дає можливість своїм користувачам швидко визначати ситуацію блокувань та швидко усувати тупикові ситуації. Блокування споживають багато системних ресурсів та значно погіршує швидкість обробки запитів до бази даних. Дуже не просто знайти ситуацію з блокуванням. Ситуація блокування: процедура зняла блокування з ресурсу А та очікує на іншу процедуру, перед цим вона буде мати доступ до В, але процедура спочатку зняла блокування з В і очікує на процедуру, що звільнить блокування на А. Для вирішення цієї ситуації необхідно швидко визначити проблему та усунути ситуацію тупику. SQL Sentry дає можливість вирішувати такі ситуації.

Вкладка «Deadlocks» показує детальну інформацію про тупики в базі даних.

«Deadlock» дає можливості [8]:

- а) Огляд операторів при виконанні запиту під час виявлення тупикової ситуації.
- б) Не встановлювати агенти та збирати інформацію про тупикові місця.
- в) Усунення перетворення SPID та ідентифікаторів посилань.
- г) Користування «Plan Explorer» для перегляду ситуацій тупиків.

З використанням «Plan Explorer» легко виявити ситуації тупиків. З використанням діаграми (кругла, за замовчуванням), можна переглядати відповідні довідкові ідентифікатори з маркуваннями.

Процеси та ресурси представляються незалежно, поряд із будь-якими зв'язками між ними [3]:

- а) Виділення червоним кольором ділянок з тупиками.
- б) Маркування цифрами та стрілками подій тупиків.
- в) Режими блокувань вказані буквами.
- г) Можливість вибрати вузли об'єктів у графічному поданні, щоб зосередити увагу на цьому об'єкті в режимі перегляду сітки.

### **1.2.2 System Center Operations Manager**

Microsoft System Center Operations Manager (SCOM) – компонент програмного забезпечення Microsoft для керування підприємствами. SCOM через єдину консоль надає можливість налаштування, керування, контролю операцій, пристроїв в межах одного підприємства. Для кожного підприємства важливо покладатися на послуги та програми для бізнесу. SCOM використовується для моніторингу та формування звітності, перевірки стану об'єктів середовища. До об'єктів середовища належить апаратний стан серверу, операційні системи та програми. Адміністратори SCOM мають можливість встановлювати, налаштовувати та користуватися об'єктами. SCOM слідкує за станом кожного об'єкту та попереджає

адміністраторів про проблеми. Також компонент пропонує можливості коригувальних дій для полегшення усунення несправностей. Кольорове відображення стану об'єктів допоможе швидко зрозуміти стан об'єкту [9]:

- Зелений, якщо проблем не виявлено.
- Жовтий для попередження.
- Червоний, якщо критична проблема.
- Сірий в разі обслуговування об'єкту, або якщо в SCOM виникають труднощі в підключенні до об'єкту.

Для надсилання попереджень адміністратори можуть встановлювати певні критерії для стану кожного об'єкту.

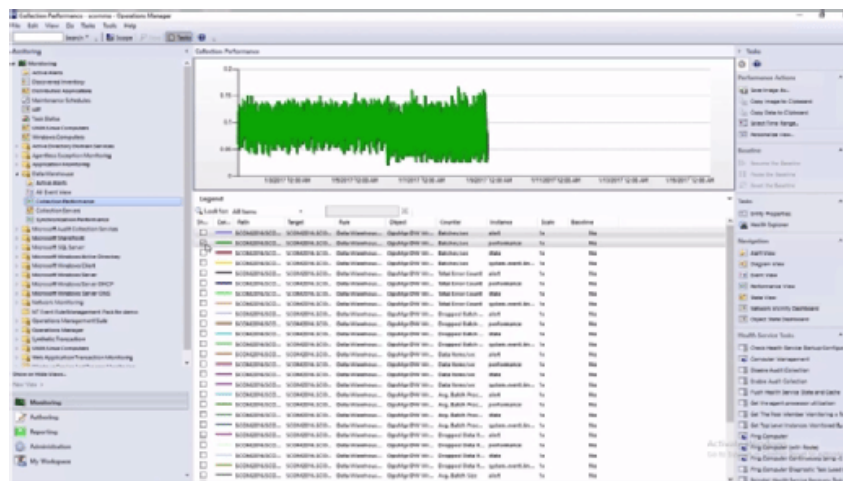


Рисунок 1.3 - Інтерфейс SCOM [9]

SCOM складається з таких компонентів:

- Сервер. Керування, адміністрування, підключення до баз даних.
- Операційна база даних. База даних для поточних звітностей.
- Сховище. Дані про SQL в звітності.

Сервер звітів звертається до бази даних і формує звіти, які будуть доставлені адміністраторам. Щоб забезпечити масштабованість, можна користуватися компонентами на одному сервері.

Для перевірки продуктивності використовуються агенти, які по замовчуванню встановлені в кожній системі. Агенти збирають та аналізують дані, які отримані сервером управління. Для збору даних та звітування певних програм, використовуються додаткові пакети. Пакети дають можливість керувати та контролювати додатками без використання безпосередньо інструментом. Використання пакетів допомагає оптимізувати процес аудиту середовища, так як процес моніторингу не є легким для користувача.

### 1.2.3 Idera DB Optimizer Product

Можливості підвищення продуктивності [10]:

- а) Перевірка продуктивності фізичних (або віртуальних) та хмарних середовищ.
- б) Перевірка запитів та планів виконання запитів для виявлення причин блокувань та взаємоблокувань.
- в) Для оптимізації продуктивності існують експертні рекомендації.
- г) Налаштування попереджувальних сповіщень, щоб уникнути помилкових повідомлень.
- г) Перегляд основних проблем та попереджень.
- д) Для аналізу повернутих запитів встановлюються критерії збору та фільтрації результатів. Зведення активних запитів по сигнатурам, збирати та відображати запити в початковому вигляді. Можливість відстеження історії виконання запиту за певний проміжок часу та виявлення запитів, які частіше всього очікували на виконання.
- е) Для інформування з проблемами порогів користувачів, можливо налаштувати сповіщення. Сповіщення автоматично відправляються користувачам на електронну пошту. Відображення сповіщень на панелі завдань операційної системи користувача, реєстрація подій в журнал (log) та ін..
- є) Визначення та усунення проблем з продуктивністю, з використанням експертних рекомендацій щодо підвищення продуктивності, включаючи виконувані сценарії. Орієнтування на деякі найбільш поширені області проблем з продуктивністю, такі як запити, конфігурація сервера, безпеку, об'єкти бази даних, пам'ять і багато іншого. За допомогою інтуїтивно зрозумілого інтерфейсу функція доступна для широкого кола користувачів [10].
- ж) Відображення, діагностика, оповіщення та звіти про критичну статистику продуктивності. Моніторинг сеансів, запитів, ресурсів, баз даних і сервісів. Збір діагностичної інформації в реальному часі з унікальною без агентною архітектурою і можливість відтворення даних [10].

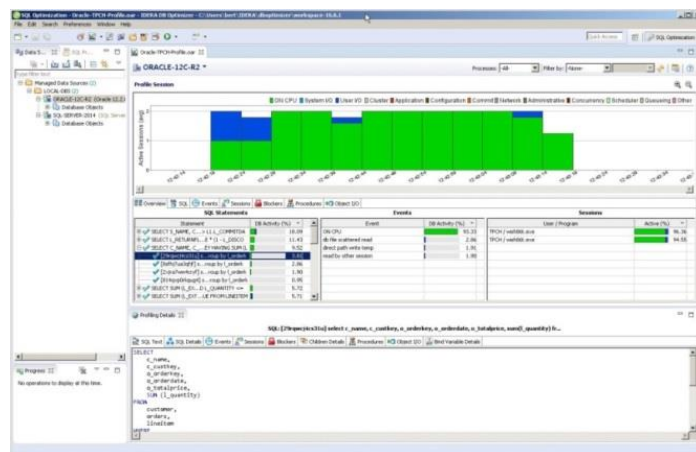


Рисунок 1.4 - Інтерфейс Idera DB Optimizer Product [10]

Оцінка виконання SQL запитів: без навантаження на базу даних, виявлення з можливістю діагностування проблемних запитів без використання агентів. Графічне відображення за допомогою діаграми таких параметрів, як:

- Використання серверного процесору.
- Введення/виведення.
- Інформація про оператори SQL.
- Статус блокувань, сеансів, процедур та ін..

В режимі реального часу можна відстежувати і інші, не дуже важливі, показники.

Генерація SQL інструкцій: завдання, які можливо створити та запустити для налаштування групи операторів (або одного). Створення та збереження підпрограм. Можливість перегляду статистики після виконання запитів.

Майстер створення SQL запиту: автоматичне пропонування рішень для налаштування SQL запиту. Графічний аналіз використання індексів, який показує проблему індексування та пропонує її оптимально налаштувати.

Візуалізація SQL запитів: перетворення SQL коду в графічні діаграми для демонстрації впливу запиту на середовище та базу даних. Відображаються індекси, статистика та об'єднання в таблицях.

Перевірка навантажень середовищ: аналіз SQL-запитів для визначення найбільш ефективного та оптимального рішення. Тестування та виконання запитів паралельно і аналіз їх впливу на середовище.

#### **1.2.4 SolarWinds**

Можливості програмного засобу для оптимізації запитів [11]:

- а) Детальний вигляд продуктивності MySQL.
- б) Машинне навчання для виявлення аномалій.
- в) Створення резервних копій та переведення середовища до останнього збереженого стану.
- г) Багатовимірний аналіз продуктивності.
- г) Ні агентів, ні впливів на продуктивність.
- д) Базові лінії, сповіщення та звіти

Аналіз продуктивності бази даних розраховує час очікувань ресурсів для виявлення ключових аспектів зниження оптимізації.

Використання інтелектуального алгоритму, який удосконалюється з часом, для визначення поведінки очікувань MySQL та MariaDB, для сповіщення відхилю від очікувань.

Для забезпечення безпеки в командах використовуються права для читання. SolarWinds

має зрозумілий веб-інтерфейс. Програмний засіб дозволяє оцінити вплив запитів на внутрішню середу.

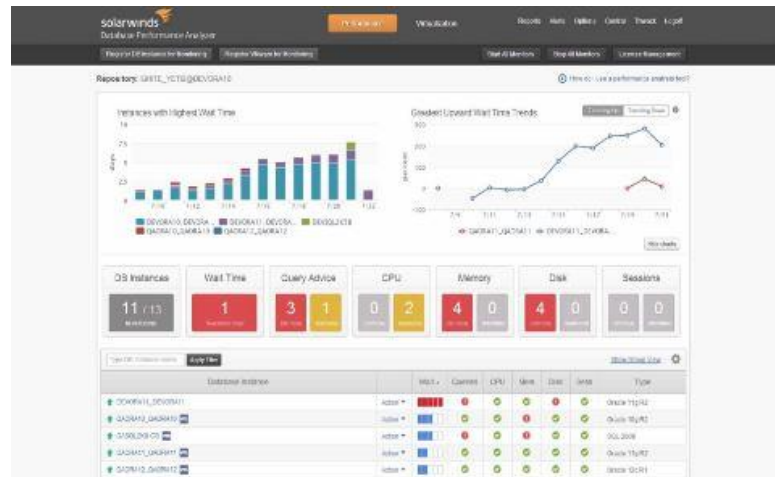


Рисунок 1.5 - Інтерфейс SolarWinds Database Performance Analyzer [11]

Для знаходження помилок та аномалій запитів, програмний засіб пропонує перегляд звітів за певний проміжок часу, в яких є дані про ефективність та метричні базові лінії.

Для обслуговування в SolarWinds немає агентів. Центральний сервер тестує всі розробки та середовище в хмарних центрах обробки даних.

Програмний засіб має змогу аналізувати запити до бази даних та їх плани виконання. Якщо після аналізу фіксуються проблеми, користувач отримує сповіщення.

Для Oracle, SAP, MySQL та ін., є підтримка на локальному рівні та з використанням віртуальних машин. В Azure та AWS зменшення витрат системних ресурсів, та можливість використовувати бази даних MySQL.

Монітор середовища фіксує та відображує ситуації, які пов'язані з блокуваннями. Для критичних запитів фіксується час очікувань. Для визначення ситуацій, які привели до проблем з очікуванням, є можливість перегляду запитів.

### 1.2.5 Quest Software

Ключові особливості [12]:

а) Моніторинг станів всіх об'єктів і вирішення знайдених аномалій та проблем. Програмний засіб підвищує ефективність перевірки стану Oracle.

б) Забезпечення підтримки всіх відомих середовищ, таких як MySQL, Oracle, SQL Server та ін., а також підтримка платформ Cloudera, Hadoop. Quest Software можливо використовувати з середовищами та не втрачати зайвий час з інтеграцією своїх проектів.

в) Програмний засіб оцінює підприємство за допомогою програмного забезпечення. При



виявленні проблем йде інформування користувачів.

г) При аномальній поведінці середовища, коли активність перевищує межі нормального стану, йде сповіщення всіх користувачів.

г) На основі зібраних даних під час моніторингу відбувається нормалізація стану середовища.

д) Зібрані дані формують звіти та представлення.

е) Програмний засіб досліджує навантаження на середовище. Це можливо за допомогою аналізу даних користувачів, процедур та сеансів.

є) Інформація за певні проміжки часу вирішує питання з тупиковими ситуаціями, блокуваннями та взаємоблокуваннями.

ж) Можливість перегляду змін середовища та бази даних.

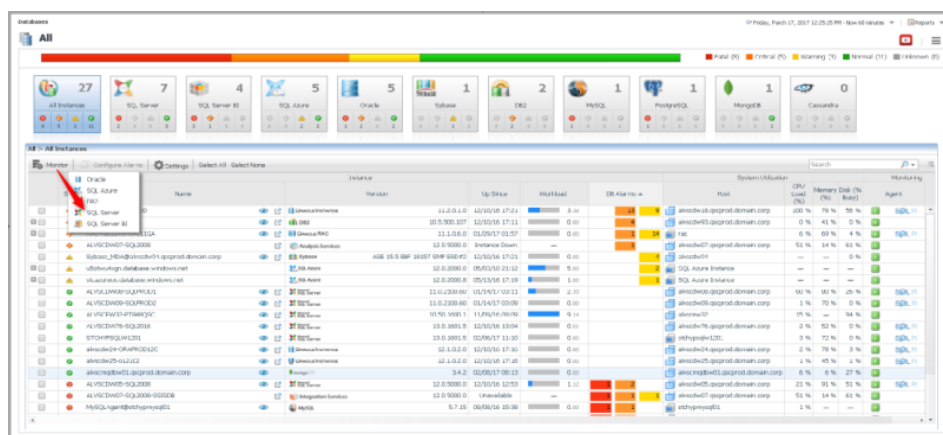


Рисунок 1.6 - Інтерфейс Quest Software [12]

## 1.2.6 Oracle Enterprise Manager

Enterprise Manger пропонує ряд властивих функцій моніторингу та управління, які в сукупності містять повноцінну систему управління середовищем. Enterprise Manager надає можливість сортувати цілі в групи для кращого керування [13].

Завдяки можливості адміністрування груп адміністратори можуть налаштувати критерії перевірки середовища, визначати стандарти шаблонування дотримань, статус життєвого циклу можливо організувати у кількох ієрархіях. Програмний додаток дає можливість своїм користувачам масштабувати ієрархію до тисяч серверів, баз даних та цільових програм.

Enterprise Manager має вбудовану систему управління, завдяки якій можливо контролювати середовище в разі складних проблем, вплив яких знижує ефективність роботи. Підтримка перегляду, зупинення та адресування подій, які відбуваються за потреби. Інтегрування ескалації інцидентів з існуючими системами підтримки.

Надання можливості визначення політики управління для адміністраторів та звичайних користувачів для автоматичного коригування ресурсів сервісу.

Enterprise Manager оптимізований для хмарних середовищ. Enterprise Manager може постійно відслідковувати велику кількість конфігурацій, щоб виявити зміни, виміряти переміщення, помилки конфігурації [13].

Пакування цілих програм, компонентів та їх публікація в хмарному сховищі. Це прискорює стадії розробки та допомагає виявити проблеми оптимізації на ранніх стадіях.

Oracle Enterprise Manager має такі основні характеристики [14]:

- а) Управління продуктивністю за допомогою автоматичного репозиторія робочого навантаження.
- б) Монітор діагностики баз даних, базових ліній AWR та адаптивних порогів.
- в) Налаштування додатків за SQL Tuning Advisor, SQL Access Advisor.
- г) Моніторинг SQL у режимі реального часу
- ґ) Моніторинг операцій з базою даних.
- д) Тестування часу відгуку за допомогою SQL Performance Analyzer.
- е) Тестування пропускну здатності.
- є) Система сповіщень про зміни в середовищі та виникнення блокувань.

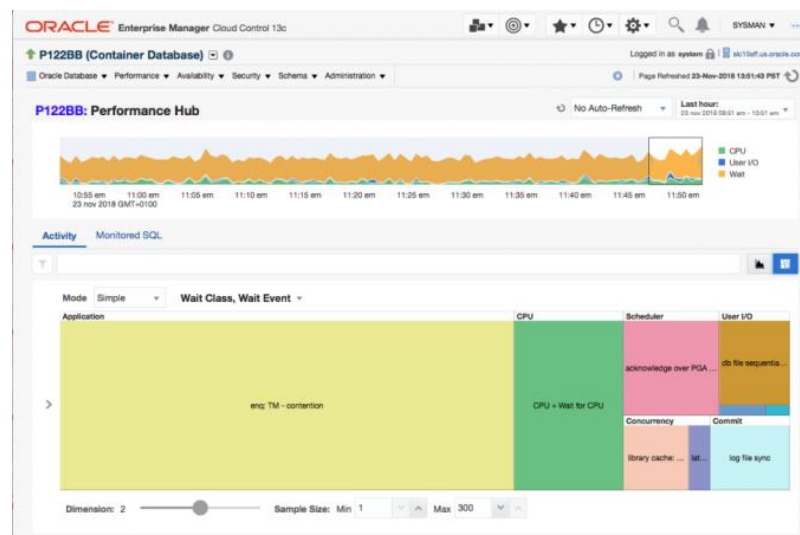


Рисунок 1.7 - Інтерфейс Oracle Enterprise Manager [14]

Проведено порівняння всіх програмних засобів (таб. 1.1), які розглядалися в цьому розділі. Порівняння відбуваються по важливим критеріям моніторингу для оптимізації бази даних. Обрано сім головних критеріїв для порівняння програмних засобів. Розглянуті засоби мають переваги та недодіки.

Після аналізу формується висновок про сенс розробки власної методики для фіксування повільних запитів.

Таблиця 1.1 – Порівняння програмних засобів.

| №  | Функція                                       | Назва програмного засобу                                  |   |   |  |  |  |
|----|---|---|---|---|--|--|--|
|    |   | SQL Sentry  | System Center Operations Manager            | Idera DB Optimizer Product                                | SolarWinds   | Quest Software                                   | Oracle Enterprise Manager OEM                    |
| 1. | Маштабовання для підтримки моніторингу        | +   | +   | невідомо  | +  | невідомо   | +  |
| 2. | Аналіз виконання SQL запиту                   | +   | +   | +   | +  | +  | +  |
| 3. | Експертні рекомендації                        | -   | -   | +   | -  | -  | +  |
| 4. | Виявлення аномалій завдяки машинному навчанню | -   | -   | -   | -  | +  | -  |
| 5. | Підтримка оптимізації MySQL, Oracle та ін..   | -   | -   | -   | +  | +  | +  |
| 6. | Наявність агенту                              | -   | +   | -   | -  | -  | -  |
| 7. | Ціна за програмний засіб                      | 14 днівна безкоштовна версія, далі ліцензія за \$2495 [8] | Ліцензія на 2 роки від \$1323 до \$3607 [9] | 14 днівна безкоштовна версія, далі ліцензія за \$572 [10] | 14 днівна безкоштовна версія, далі ліцензія за \$2045 [11] | 14 днівна безкоштовна версія, далі невідомо [12] | 14 днівна безкоштовна версія, далі невідомо [14] |

### 1.3 Постановка наукової задачі та обґрунтування методики досліджень

Результати проведеного аналізу моделей, методів й інструментальних засобів оптимізації запитів до бази даних показали, що у відомих публікаціях не вирішеною є задача вибору основного методу для оптимізації запитів до бази даних.

У відомій літературі «Програмування баз даних SQL» [15], «High Performance MySQL» [16] та «Refactoring SQL Applications» [17] задачі, пов'язані з моделюванням обробки великої кількості даних при запиті до бази даних не розглядалися. Не розв'язувалися також задачі моніторингу стану бази даних під час блокувань, тупиків та взаємоблокувань. Розглядалася структура SQL запиту, оператори запитів та їх вплив на базу даних. Також йшла мова про те, що вмiле проектування схеми бази даних дуже важливе в подальшому та значно покращує продуктивність бази даних. Розглядалися критерії, які оцінювали якість SQL запиту.

В даному контексті слід визначити задачі магістерської роботи:

- 1) Аналіз існуючих програмних засобів для моніторингу стану бази даних
- 2) Вибір критичних метрик БД.
- 3) Встановлення порогових значень для обраних метрик.
- 4) Вибір математичного методу для розрахунку ймовірностей оптимізації.
- 5) Виявлення оптимального методу оптимізації запитів.
- 5) Розробка методики, яка б дозволила поліпшити швидкість виконання запитів до бази даних.
- 6) Проведення експерименту для оцінки якості методу оптимізації запитів.

## Висновки до першого розділу

У першому розділі були розглянуті такі програмні засоби (ПЗ), як: SQL Sentry, System Center Operation Manager, Idera DB Optimizer Product, Quest Software, Oracle Enterprise Manager. Всі ці ПЗ спрямовані на моніторинг станів бази даних. Моніторинг є корисним засобом, завдяки якому можна знайти проблеми на ранніх стадіях і швидко їх вирішити. Серед переваг використання ПЗ можна виділити такі функції:

- Масштабування для підтримки моніторингу.
- Аналіз виконання SQL запиту.
- Експертні рекомендації.
- Виявлення аномалій завдяки машинному навчанню.
- Підтримка оптимізації.

З використанням ПЗ можуть виникнути труднощі з несвоєчасним реагування на критичні стани бази даних/середовища, малою кількістю інформації для прийняття рішення, виявленні аномалій.

Несвоєчасне реагування на критичні стани бази даних/середовища зменшує продуктивність бази даних та є загроза втрати даних. В режимі реального часу немає можливості швидко впливати на ситуацію та вирішувати критичні проблеми.

Більшість ПЗ не надають повної інформації про виникнення проблем в базі даних. Звіти про проблеми не завжди можуть надати повне уявлення про проблему та її першопричини. Важливо розуміти всі причини виникнення проблеми та можливість в майбутньому її відстежити/вирішити на основі отриманої інформації.

Програмні засоби не визначають відхилення параметрів від їх стану по замовчуванням. В разі виникнення аномалій, їх не можна буде виявити та виправити.

Було прийнято рішення про створення механізму для своєчасного оповіщення та надання інформації для оптимізації запиту до бази даних. Механізм не буде мати тих недоліків, які є в розглянутих ПЗ.

## Література до першого розділу

1. William Charles Eidson, Jesse Collins. Methods and systems for joining indexes for query optimization in a multi-tenant database. United States Patent – Режим доступу: [www. URL:   
https://patents.google.com/patent/US9405797B2/en](http://www.patents.google.com/patent/US9405797B2/en) - Aug. 2, 2016.
2. Edward Gust BRANISH, John F. Hornibrook, Dieu Quang, La Calisto, Paul Zuzarte. Optimization of aggregate queries in database management systems using an early out join when processing min and max functions. United States Patent – Режим доступу: [www. URL:   
https://patents.google.com/patent/US9524317B2/en](http://www.patents.google.com/patent/US9524317B2/en) - Dec. 20, 2016.
3. Xing Chen, Qing Yun Hao, Yi Jin, Wan Chuan Zhang. Database query in a share-nothing database architecture». United States Patent – Режим доступу: [www. URL:   
https://patents.google.com/patent/US9146959B2/en](http://www.patents.google.com/patent/US9146959B2/en) - 2015.
4. ЗБІРНИК НАУКОВО-ПРАКТИЧНИХ ПРАЦЬ ІV МОЛОДІЖНОГО ФОРУМУ «ІТ-ІДЕЯ 2018» (07.12.2018 р.) / видавництво Східноукраїнського національного університету ім. В. Даля. – Сєвєродонецьк: 2018. – 44-45 с.
5. Майбутній науковець – 2018 : матеріали всеукр. наук.-практ. конф. з міжнар. участю (14 груд. 2018 р.), м. Сєвєродонецьк. Ч. ІІ / укладач В. Ю. Тарасов – Сєвєродонецьк : Східноукр. нац. ун-т ім. В. Даля, 2018. – 28-29 с.
6. Query Optimization in Database Systems: III International Conference TACSIT-2019 Proceedings, May 7-8, 2019, Severodonetsk, Ukraine / I. Skarga-Bandurova (Ed.). – Severodonetsk: Volodymyr Dahl East Ukrainian National University, 2019. – 35-36 с.
7. DATABASE QUERY OPTIMIZATION: ВІСНИК СХІДНОУКРАЇНСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ імені Володимира Даля № 5 (253) 2019 / Відповідальний за випуск: д.т.н., професор Скарґа-Бандурова І.С – Сєвєродонецьк: СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Володимира Даля, 2019. – 78-83 с.
8. SQL Sentry: Breakthrough SQL Server Performance Monitoring – Режим доступу: [www. URL:   
https://www.sentryone.com/products/sentryone-platform/sql-sentry/sql-server-performance-monitoring](http://www.sentryone.com/products/sentryone-platform/sql-sentry/sql-server-performance-monitoring)
9. Microsoft Ignite 2018 conference coverage. Microsoft SCOM (System Center Operations Manager) – Режим доступу: [www. URL:   
https://searchwindowsserver.techtarget.com/definition/Microsoft-System-Center-Operations-Manager-Microsoft-SCOM](http://www.searchwindowsserver.techtarget.com/definition/Microsoft-System-Center-Operations-Manager-Microsoft-SCOM)
10. IDERA. DB Optimizer Product Tour - Режим доступу: [www. URL:   
https://www.idera.com/dboptimizer-sql-database-optimization/product-tour](http://www.idera.com/dboptimizer-sql-database-optimization/product-tour)
11. solarwinds. Database Performance Analyzer for MySQL - Режим доступу: [www. URL:   
https://www.solarwinds.com/database-performance-analyzer-mysql](http://www.solarwinds.com/database-performance-analyzer-mysql)

12. Quest. Database performance monitoring - Режим доступа: [www. URL: https://www.quest.com/solutions/database-performance-monitoring/](http://www.quest.com/solutions/database-performance-monitoring/)
13. Oracle® Enterprise Manager. Cloud Administration Guide [Электронный ресурс] / Oracle Enterprise Manager Cloud Administration Guide 12c Release 5 (12.1.0.5) - Режим доступа: [www. URL: https://docs.oracle.com/cd/E24628\\_01/doc.121/e28814.pdf](http://www.oracle.com/docs/cd/E24628_01/doc.121/e28814.pdf) - грудень, 2015 р.
14. Oracle. Enterprise Manager Database Management - Режим доступа: [www. URL: https://www.oracle.com/database/technologies/managability.html](http://www.oracle.com/database/technologies/managability.html)
15. Карвин Б. Программирование баз данных SQL. Типичные ошибки и их устранение / Карвин Б – М.: Рид Групп, 2012, 336 с.
16. High Performance MySQL / Baron Schwartz, Peter Zaitsev, Vadim Tkachenko, Jeremy D. Zawodny, Arjen Lentz, Derek J. Balling: O`REILLY, 2010, 823 с.
17. Refactoring SQL Applications / Stephane Faroult with Pascal L`Hermite: O`REILLY, 2009, 32 с.

## РОЗДІЛ 2

### АНАЛІЗ ПРОБЛЕМ ОПТИМІЗАЦІЇ ЗАПИТІВ

#### 2.1 Метрики для підвищення продуктивності

Оптимізація запитів дуже важлива для роботи системи. Для зведення до мінімумів використання системних ресурсів серверу та для прискорення часу виконання запитів використовуються засоби оптимізації. Час виконання запитів є показником продуктивності баз даних. Під час виконання запиту необхідно виявити джерело сповільнення часу обробки. Джерелом сповільнення запиту може бути проблема в базі даних або в структурі запиту. Необхідно обрати метрики для моніторингу стану бази даних та обрати декілька з них. Показники обраних метрик зафіксують проблему на боці серверу. Проблему структури запиту метрики не зафіксують. При проектуванні структури бази даних необхідно спланувати структуру для мінімізації часу пошуку необхідних даних. З урахуванням методів оптимізації, час очікування інформації мінімізується та велика кількість користувачів зменшує навантаження на сервер та використовується значно менша кількість системних ресурсів. Сканувати мільйони, мільярди чи трильйони рядків, щоб повернути лише два-три - це величезна трата ресурсів [1].

Метрики фіксують значення, що відносяться до систем у певний момент часу [2].

Для виявлення проблем з базою даних можна користуватися такими метриками:

- а) Показники навантаження на центральний процесор (ЦП) серверу.
- б) Показники кількості операцій зчитування/запису з запам'ятовуючого пристрою серверу.
- в) Показниками навантаження на оперативну пам'ять (RAM) серверу.
- г) Значення об'єму буферного кешу.
- д) Значення швидкості мережі.
- е) Значення вільної фізичної пам'яті.

В таблиці 2.1 приведені метрики, завдяки яким можна відстежити стан бази даних під час виконання запиту та на основі їх показників виявити проблему продуктивності. Для моніторингу стану бази даних існують багато метрик, на основі яких можна приймати рішення про поточний стан.

Метрики ресурсів сигналізують про навантаження центрального процесору, операції зчитування/запису, навантаження на оперативну пам'ять (RAM) та на очікування (waits) бази даних. Метрики показників ефективності (performance satisfaction metrics) сигналізують про час обробки запитів в секунду, транзакції в секунду, пропускну здатність, час обробки запиту.



Таблиця 2.1 – Метрики для відстеження продуктивності бази даних.

| №  | Метрика  | Призначення  |
|----|--|--|
| 1  | CPU [3]  | Відстеження стану центрального процесору (ЦП) серверу.   |
| 2  | Storage I/O [3]  | Відстеження операцій зчитування/запису.  |
| 3  | RAM [3]  | Використання пам'яті для обробки запиту.   |
| 4  | Global Cache [3]   | В випадках ідентичних запитів повертати відповідь із кешу для зберігання ресурсів та повторно не виконувати ідентичні запити.                        |
| 5  | Response time [3]  | Час реакції бази даних на запити та їх обробку.  |
| 6  | Transactions per second [3]                                    | Кількість транзакцій в секунду.  |
| 7  | Client/server bytes received by server per sec [4]             | Кількість байтів клієнта/сервера, отриманих сервером за секунду.   |
| 8  | Deadlocks per minute [4]                                       | Кількість блокувань (тупиків) в секунду  |
| 9  | Highest amount of permanent partition in use – MB (MEMORY) [4] | Відсоток постійного простору розділів, який використовується.  |
| 10 | Lock Wait Duration (s) [5]                                     | Час, в секундах, скільки агент чекав для доступу.  |
| 11 | Physical writes [6]  | Загальна кількість блоків даних, записаних на диск. Це значення статистики дорівнює сумі фізичних записів прямих і фізичних записів із значень кешу. |

Використовувати всі метрики з таблиці 2.1 для підвищення оптимізації запиту не обов'язково. За допомогою наукової літератури необхідно проаналізувати вплив деяких метрик на базу даних та обрати лише ті, які більше всього впливають на продуктивність.

В статті [7] розглядаються показники кількості операцій зчитування/запису та кількість використаних ресурсів центрального процесору.

В патенті [8] основною метрикою для аналізу стану баз даних є кількість зчитувань/записів.

Однією з основних метрик для аналізу є пропускна здатність системи та час її відгуку. З урахуванням двох чи більше факторів, отримані дані будуть відображати реальний стан системи в окремо взятій проміжок часу. З використанням цих метрик є сенс аналізувати поточні стани системи не залежно від їх структури, класу та вимог класу. Для систем управління баз даних це є час очікування на запит користувачів та час пошуку необхідної інформації. Необхідно фіксувати значення показників, якщо час обробки запитів збільшується. З урахуванням середньої кількості дій розраховується пропускна здатність системи, яка є одним

з параметрів продуктивності бази даних. Час відгуку є величиною продуктивності, яка визначає середній час очікування на результати запиту [9].

Процесори можуть ділити навантаження на сервер, що призводить до підвищення продуктивності. На ефективність роботи з базою даних залежить не тільки база даних, але також впливають й інші процеси, що працюють на сервері, тому важливо переглянути загальне навантаження, а також використання бази даних. Оскільки використання процесора змінюється протягом дня, показники повинні бути вивчені на періоди низького використання, середнього використання та пікового використання, щоб найкраще оцінити, чи будуть корисні додаткові ресурси процесора.

Обмежена або погано виділена фізична пам'ять впливає на продуктивність бази даних. Чим більше пам'яті доступно, тим кращою буде продуктивність.

Буфер кеш-пам'яті покращує продуктивність, зберігаючи дані в пам'яті ядра та усуваючи зчитування/запис диску. Якщо кеш-пам'ять занадто мала, дані видаляються з кешу частіше. Якщо вона потрібна ще раз, її потрібно зчитувати з диска. Окрім повільності зчитування диска, це додає додаткових робіт на пристрої зчитування/запису і може стати вузьким місцем, Налаштовані SQL-запити можуть допомогти більш ефективно використовувати кеш-буфер.

Швидкість мережі може впливати на швидкість повернення отриманих даних кінцевому користувачеві або процес виклику. Необхідно контролювати якість підключення до віддалених баз даних.

Серед основних метрик для контролю стану бази даних на боці серверу будуть використовуватися метрики:

- 1) Навантаження на центральний процесор.
- 2) Навантаження на пам'ять.
- 3) Зчитування/запис запам'ятовуючого пристрою.

## **2.2 Виявлення трудомістких SQL запитів**

Для відстеження запущених запитів та виявлення проблемних з них існують такі методи:

- 1) SHOW PROCESSLIST [10].
- 2) Query Log [11].

Методи необхідні для відстеження станів бази даних під час виконання запитів та виявленні проблем. Проблемні запити сповільнюють обробку запитів та зменшують продуктивність бази даних. Кожен з методів відображає інформацію, яка допомагає виявити та вирішити проблему. Методи різняться між собою, але мають єдину мету – відстеження запущених SQL-запитів.

## 2.2.1 SHOW PROCESSLIST

SHOW PROCESSLIST показує, які потоки запущені. Якщо є привілей PROCESS, можна побачити всі потоки. В іншому випадку можна бачити лише власні потоки. Якщо не використовується ключове слово «FULL», у полі «Інформація» відображаються лише перші 100 символів кожного твердження.

Оператор SHOW PROCESSLIST дуже корисний, якщо отримується повідомлення про помилку "занадто багато з'єднань" і необхідно дізнатися, що відбувається. MySQL залишає ще одне додаткове підключення для використання обліковими записами, які мають привілей CONNECTION\_ADMIN або SUPER, щоб адміністратори завжди мали змогу підключитись та перевіряти систему (якщо вважати, що ви не надаєте цю привілей усім своїм користувачам).

У стовпці "Інформація" відображається виконаний запит або NULL, якщо в даний час нічого не відбувається. При запуску "Показувати список процесів" він відобразить лише перші 100 символів запиту. Щоб показати повний запуск запиту, натомість "показати повний список процесів". Запуск вищевказаної команди з інтерфейсу командного рядка MySQL. Розмежувач «а» може ускладнити зчитування результатів, особливо якщо запити довгі та охоплюють кілька рядків. Використання обмежувача \ G натомість покаже дані у форматі, який часто читається, хоча він займає більше рядків у терміналі. Це особливо корисно при запуску "показати повний список процесів", оскільки деякі запити можуть бути досить довгими [12].

```
mysql> SHOW PROCESSLIST;
```

| Id | User | Host      | db   | Command | Time | State | Info             |
|----|------|-----------|------|---------|------|-------|------------------|
| 10 | root | localhost | NULL | Query   | 0    | init  | SHOW PROCESSLIST |
| 11 | root | localhost | NULL | Sleep   | 642  |       | NULL             |
| 12 | root | localhost | NULL | Sleep   | 624  |       | NULL             |
| 13 | root | localhost | NULL | Sleep   | 621  |       | NULL             |
| 14 | root | localhost | NULL | Sleep   | 617  |       | NULL             |
| 16 | root | localhost | NULL | Sleep   | 555  |       | NULL             |
| 17 | root | localhost | NULL | Sleep   | 552  |       | NULL             |
| 19 | root | localhost | NULL | Sleep   | 16   |       | NULL             |
| 20 | root | localhost | NULL | Sleep   | 14   |       | NULL             |
| 21 | root | localhost | NULL | Sleep   | 11   |       | NULL             |
| 22 | root | localhost | NULL | Sleep   | 8    |       | NULL             |

11 rows in set (0.00 sec)

Рисунок 2.1 – результати SHOW PROCESSLIST [13]

SHOW PROCESSLIST має такі стовпці [12]:

- ID. Ідентифікатор з'єднання. Це той самий тип значення, який відображається в стовпці ідентифікатора таблиці INFORMATION\_SCHEMA PROCESSLIST, стовпчику PROCESSLIST\_ID таблиці потоків схеми продуктивності.

- User. Користувач MySQL, який видав заяву. Це може бути потік вводу/виводу або SQL, який використовується у ведених реплікаціях або обробник із затримкою рядка.

- Host. Ім'я хоста клієнта, який видає заяву (за винятком користувача системи, для якого немає хоста). Ім'я хоста для підключень TCP/IP повідомляється у форматі host\_name: client\_port.

- DB. База даних за замовчуванням, якщо вибрана така; в іншому випадку NULL.

- Command. Тип команди, яку виконується.

- Time. Час у секундах, коли потік перебуває у своєму поточному стані. Для підлеглого потоку SQL значення - це кількість секунд між часовою позначкою останнього повтореного події та реальним часом.
- State. Дія, подія або стан, що вказує, що обробляє потік.
- Info. Оператор, який виконується потоком, або NULL, якщо він не виконує жодного оператора.

### 2.2.2 Query Log

Query Log - записи про фіксування роботи серверу під час обробки запитів. Сервер зберігає інформацію в журнал під час використання базою даних користувачами та реєстрування кожного запиту, отриманого від клієнтів. Головною перевагою методу є фіксування помилок SQL запитів клієнтів та надання інформації про кожну з помилок. Надана інформація допомагає вирішити проблему та прискорити продуктивність бази даних.

Кожне з'єднання клієнту з базою даних фіксується та визначається тип протоколу з'єднання.

Запис записів в журнал відбувається в порядку їх отримання на сервер та може відрізнитися від порядку їх виконання. Порядок ведення журналу відрізняється від двійкового журналу. В двійковий журнал оператори записуються після обробки, але без урахувань блокувань.

За замовчуванням Query Log вимкнено. Для його активації (Maria BD та ін..) необхідно встановити системну змінну `General_log` на 1.

Загальний журнал запитів виводиться у файл за замовчуванням. Однак його можна явно вибрати, встановивши змінну системи `log_output` на FILE.

Якщо загальний файл журналу запитів увімкнено, але ім'я не вказано, ім'я за замовчуванням - `host_name.log`, і сервер створює файл у тому самому каталозі, де створюється файл PID. Якщо ім'я введено, сервер створює файл у каталозі даних, якщо не вказано абсолютне ім'я шляху, щоб вказати інший каталог.

```
36 Query insert questions_c23(qId,ownerId,title,votes,answers,isClosed,closeVotes,views,ownerRep,
comments,answeredAccepted,askDate,closeDate,lastScanDate,ign,bn,pvtc,
mainTagForImport,prepStatus,touches,status,status_bef_change,cv_bef_change,max_cv_reached)
values(38666373, 1322183, 'How to post a numeric value in c#', 0, 1, 0, 0, 50, 1,
0, 0, '2016-07-29 19:40:32', null, now(), 0, 0, 0,
'c%23',0,1,'0','',0,0)
on duplicate key update title='How to post a numeric value in c#', votes=0, answers=1, views=50,ownerRep=1,
answeredAccepted=0,lastScanDate=now(), touches=touches+1,status='0'
```

Рисунок 2.2 – фрагмент Query Log [14]

«SHOW PROCESSLIST» надає більше інформації про проблемні запити, ніж «Query Log». «SHOW PROCESSLIST» показує користувачів бази даних, які виконують запити в реальному часі, які саме запити виконують та скільки саме часу було необхідно для виконання

запитів, а «Query Log» лише реєструє проблемні запити. Для відстеження станів запитів було обрано «SHOW PROCESSLIST».

### 2.3 Виявлення повільного запиту та вибір математичного методу

За допомогою журналу повільних запитів (slow query log file) було виявлено повільний запит, час його обробки, час блокувань та кількість перевірених рядків під час його виконання. Повільний запит – пошук інформації про навантаження по курсам. Було виявлено 89 подібних запитів.

```

Time                Id Command      Argument
# Time: 191003 21:31:18
# User@Host: nygma[nygma] @ localhost [127.0.0.1]
# Query_time: 11.574986  Lock_time: 0.001999 Rows_sent: 1  Rows_examined: 1696607
use db610668427z;
SET timestamp=1570127478;
SELECT ID_CS_SEMESTR, MAX(LECTION) AS LECTION, MAX(PRACTICAL) AS PRACTICAL,
MAX(LABS) AS LABS, SM_STATE, CP, CW, EXAMS, TEST,SEMESTR_V FROM cs_v_semestr WHERE ID_C_CP IN
(SELECT ID_C_CP FROM full_int_v_plans WHERE YEAR_PL= 2017 AND ID_COURSE = 53
AND LEARN_TIME=1 GROUP BY YEAR_PL, SEMESTR_V, ID_COURSE) AND SEMESTR_V IN (2,4,6) GROUP BY SEMESTR_V;

```

Рисунок 2.3 - Фрагмент журналу повільних запитів

Кількість зчитувань інформації впливає на показники навантаження процесору [19]. Процесор очікує зчитувальні дані та перебуває в простій до тих пір, поки не обробить дані, потім знову очікує дані, які досить повільно зчитуються. Це впливає на продуктивність системи, яка не перебуває в своїй максимальній продуктивності. Метрики, за якими треба стежити при виконанні подібних запитів – зчитування/запис та навантаження процесору серверу.

За допомогою журналу повільних запитів (slow query log file) було виявлено повільний запит, час його обробки, час блокувань та кількість перевірених рядків під час його виконання.

Повільний запит – пошук інформації про навантаження по курсам. Було виявлено 89 подібних запитів.

Для розрахунку ймовірностей та аналізу отриманих даних для метрик можна скористатися методом Демпстера-Шафера (DST) [18] або Дезерта-Смарандаша (DSmT) [22].

Детальний алгоритм дій для виявлення проблем обробки запитів та методів оптимізації.

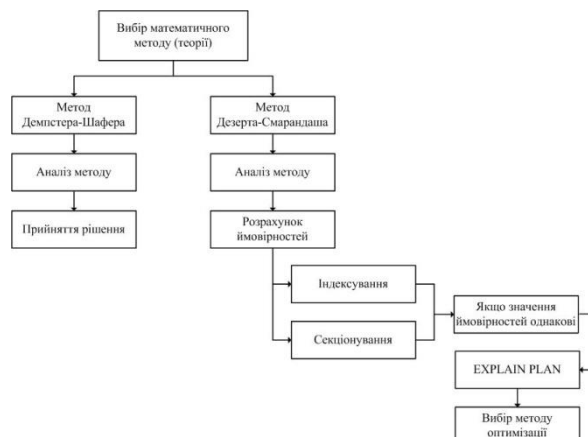


Рисунок 2.4 - Схема виявлення проблем обробки запитів та методів оптимізації

Теорія Демпстера-Шафера базується на двох ідеях: отримання ступеня переконання на одне питання від суб'єктивних ймовірностей для пов'язаного питання та правила Демпстера для поєднання таких ступенів віри, коли вони засновані щодо незалежних доказів. Ступінь віри в пропозицію залежить насамперед від кількості відповідей (на відповідні питання), що містять пропозицію, та суб'єктивної вірогідності кожної відповіді. Також сприяють правила поєднання, які відображають загальні припущення щодо даних. Теорія Демпстера-Шафера призначає свої маси всім підмножинам пропозицій, що складають систему - в теоретико-теоретичному відношенні, потужність набору пропозицій.

Проаналізувавши метод Демпстера-Шафера, можна зробити висновок, що:

- Правило Демпстера інтерпретується як метод кумулятивного об'єднання переконань, слід очікувати добре відомих суперечливих результатів, отриманих в результаті об'єднання суперечливих переконань. У разі низького конфлікту між двома аргументної переконаннями, правило Демпстера є наближення кумулятивного злиття переконань, але в міру збільшення рівня конфлікту точність апроксимації зменшується. Навіть в разі низького конфлікту між двома аргументної переконаннями правило Демпстера, якщо його інтерпретувати як метод кумулятивного об'єднання переконань, може привести до досить суперечливих результатів [20].

- Правило комбінування Демпстера насправді є методом об'єднання обмежень переконань. Він представляє тільки приблизний оператор злиття в інших ситуаціях, таких як кумулятивне злиття переконань, але зазвичай дає невірні результати в таких ситуаціях. Таким чином, плутанина навколо обґрунтованості правила Демпстера виникає через неправильну інтерпретацію природи ситуацій, які необхідно змодельювати [20].

Для отримання більш точних даних було прийнято рішення використовувати теорію Дезерта-Смарандаша.

Теорія Дезерта-Смарандаша (DSmT) правдоподібних і парадоксальних міркувань є продовженням класичної теорії Демпстера-Шафера (DST), але включає фундаментальні відмінності з DST. DSmT дозволяє формально поєднувати будь-які типи незалежних джерел інформації, представлених функціями переконань, але в основному орієнтований на злиття невизначених, сильно суперечливих та неточних кількісних чи якісних джерел доказів. DSmT здатний вирішувати складні, статичні або динамічні задачі синтезу поза межами рамки DST, особливо коли конфлікти між джерелами стають великими і коли уточнення кадру розглянутої проблеми стає недоступним через невиразний, відносний та неточний характер [19].

За допомогою DmST буде виконано розрахунок ймовірностей метрик (CPU, I/O) та буде виконано злиття даних для виявлення необхідності оптимізації для запитів.

### 2.3.1 Розрахунок показників метрик

Для розрахунку ймовірності навантаження на кількість операцій зчитування/запису необхідно визначити кількість зчитувань та загальну кількість їх повторів.

В таблиці 2.2 приведена загальна кількість зчитувань та кількість повторів зчитувань для кожного з випадків зчитування даних.

Таблиця 2.2 – Кількість зчитувань та їх повторів.

| Зчитування | Повтори зчитувань |
|------------|-------------------|
| 1694417    | 32                |
| 1696608    | 23                |
| 2897825    | 8                 |
| 2900016    | 7                 |
| 1694419    | 5                 |
| 1983528    | 4                 |

Згідно з таблицею 2.2 необхідно визначитись з найбільшою кількістю зчитувань та повторів цих зчитувань [15]. Необхідно проаналізувати кожний стан та визначити кількість використаних системних ресурсів. Обирається змінна стану бази (Last\_query\_cost). З урахуванням отриманих значень розраховуються ймовірності.

```
| Last_query_cost | 3499.799000 |
```

Рисунок 2.5 - Значення системних ресурсів для значення зчитування 1694416

```
|| Last_query_cost || 2025.563064 |
```

Рисунок 2.6 - Значення системних ресурсів для значення зчитування 1696607

```
| Last_query_cost | 33.637641 |
```

Рисунок 2.7 - Значення системних ресурсів для значення зчитування 2897824

```
| Last_query_cost | 26.431503 |
```

Рисунок 2.8 - Значення системних ресурсів для значення зчитування 2900015

Виявлені два стани зчитування даних, на обробку яких необхідно найбільше всього системних ресурсів.

Згідно з формули 2.1 розраховується ймовірність навантаження на зчитування/запис [21].

$$P(A) = \frac{m}{n} \quad (2.1)$$

де  $n$  – значення повторення станів,  $m$  – значення кількості повторів станів, на обробку яких використовується велике значення системних ресурсів.

Розрахунок ймовірності навантаження на зчитування/запис даних.

$$P_{\max}(A) = \frac{55}{79} = 0.69$$

де 55 – кількість зчитувань з використанням найбільших значень системних ресурсів.

Розрахована ймовірність навантаження на кількість зчитувань/запису даних під час обробки запиту становить 0.69. Розрахунок мінімальної ймовірності використання системних ресурсів для операції зчитування/запису даних.

$$P_{\min}(A) = \frac{15}{79} = 0.19$$

Порогове значення (threshold) для зчитування/запису є значення навантаження 69%.

З використанням `slow query log` виявлені навантаження на центральний процесор під час обробки запиту. Всі стани навантаження приведені в таблиці 2.3. Таблиця містить стани навантажень та їх загальну кількість для розрахунку ймовірностей навантаження на центральний процесор.

Таблиця 2.3 – Значення навантажень центрального процесору та їх повторів

| Значення навантажень, % | Повтори |
|-------------------------|---------|
| 40                      | 110     |
| 51                      | 94      |
| 76                      | 86      |
| 49                      | 72      |
| 68                      | 70      |
| 22                      | 63      |
| 47                      | 54      |
| 63                      | 47      |

Згідно з таблиці 2.4, діапазон значень навантажень на центрального процесору становить від 22% до 76%. В отриманому діапазоні даних необхідно обрати значення, які негативно впливали на продуктивність бази даних. Обрані стани:

а) 76.



б) 68.

в) 63.

Мінімальне значення навантаження центрального процесору – 22%.

Порогове значення (threshold) для центрального процесору є значення навантаження 76%. В таблиці 2.5 приведені порогові значення для зчитування/запису та навантаження на центральний процесор. Коли навантаження перевищує ці значення, необхідно сповіщати про це користувача бази даних.

Розрахунок ймовірності максимального навантаження на центральний процесор з урахуванням кількості повторів станів та діленням на загальну кількість значень

$$P_{\max}(A) = \frac{m}{n} = \frac{203}{596} = 0.34$$

Розрахунок мінімальної ймовірності навантаження на центральний процесор [21].

$$P_{\min}(A) = \frac{63}{596} = 0.1$$

В таблиці 2.4 приведені всі розраховані ймовірності розглянутих метрик.

Таблиця 2.4 – Розраховані ймовірності для обраних метрик.

| Параметр             | Максимальна ймовірність | Мінімальна ймовірність |
|----------------------|-------------------------|------------------------|
| Зчитування/запис     | 0.69                    | 0.19                   |
| Центральний процесор | 0.34                    | 0.1                    |

Для розрахунку порогових значень для обраних метрик необхідно визначити середнє арифметичне значення від розрахованих ймовірностей. Розраховані порогові значення ймовірностей метрик приведені в таблиці 2.5.

Таблиця 2.5 – Розрахункові значення порогових значень для обраних метрик.

| Метрика стану        | Значення |
|----------------------|----------|
| Зчитування/запис     | 0.44     |
| Центральний процесор | 0.22     |

За допомогою обраної теорії Дезерта-Смарандаша (DSmT) необхідно зробити злиття ймовірностей метрик для того, щоб не відстежувати більше показники навантажень.

Групування ймовірностей в загальну множину [36]:

$$2^{\Theta=\{X_1, X_2\}} = \{\emptyset, X_1, X_2, X_1 \cup X_2\} = \{\emptyset, \{X_1\}, \{X_2\}, \{X_1 \cup X_2\}\}$$

Загальна множина для розглянутих ймовірностей [30]:

$$2^{\Theta=\{A,B\}} = \{\emptyset, \{0.69, 0.19\}, \{0.34, 0.1\}, \{0.69, 0.19\}, \{0.34, 0.1\}\}$$

У DST поєднання двох незалежних джерел доказів характеризується  $m1(.)$  і  $m2(.)$  робиться за допомогою правила Демпстера наступним чином [16]:

$$M_{DS}(X) = \frac{\sum_{\substack{x_1, x_2 \in 2^x \\ x_1 \cap x_2 = X}} m_1(x_1)m_2(x_2)}{1 - \sum_{\substack{x_1, x_2 \in 2^x \\ x_1 \cap x_2 = \emptyset}} m_1(x_1)m_2(x_2)} \quad (2.2)$$

Правило про перерозподіл пропорційного конфлікту (PCR5) пропонується як серйозна альтернатива правилу DST [22]:

$$m_{PCR5} = \sum_{\substack{x_1 \in 2^x \\ x_1 \cap x_2 = X}} m_1(X_1)m_2(X_2) + \sum_{\substack{x_2 \in 2^x \\ x_2 \cap x_1 = \emptyset}} \left[ \frac{m_1(X_1)^2 m_2(X_2)}{m_1(X_1) + m_2(X_2)} + \frac{m_2(X_1)^2 m_1(X_2)}{m_2(X_1) + m_1(X_2)} \right] \quad (2.3)$$

З використанням обраного математичного методу DSmT, розраховуються ймовірності показників навантажень для метрик. Математичний розрахунок з використанням DSmT відображає стан бази в поточний стан бази даних.

З використанням формули 2.2, злиття ймовірностей навантаження зчитування/запису (Disk I/O) по правилу DST становить:

$$M_{DS}(X) = \frac{0.69 * 0.34}{1 - (0.69 * 0.34)} = \frac{0.2346}{0.7654} = 0.306$$

З використанням формули 2.3, розрахунок ймовірності навантаження на зчитування/запис з використанням DSmT

$$M_{PCR5}(X) = 0.69 * 0.34 + \left[ \frac{0.69^2 * 0.34}{0.69 + 0.34} + \frac{0.34^2 * 0.69}{0.34 + 0.69} \right] = 0.4676$$

З використанням формули 2.2, розрахунок ймовірності навантаження на центральний процесор з використанням DST:

$$M_{DS}(X) = \frac{0.34 * 0.19}{1 - (0.34 * 0.19)} = \frac{0.0646}{0.9354} = 0.069$$

Згідно з формули 2.3, розрахунок ймовірності навантаження на центральний процесор з використанням DSmT:

$$M_{PCR5}(X) = 0.34 * 0.19 + \left[ \frac{0.34^2 * 0.19}{0.34 + 0.19} + \frac{0.19^2 * 0.34}{0.19 + 0.34} \right] = 0.1256$$

В таблиці 2.6 наведені розрахунки злиття показників по правилах DST та DSmT. Розраховані значення різні, що свідчить про те, що DST метод представляє тільки приблизний оператор злиття в інших ситуаціях, таких як кумулятивне злиття переконань, але зазвичай дає невірні результати в таких ситуаціях. Таким чином, плутанина навколо обґрунтованості правила Демпстера виникає через неправильну інтерпретацію природи ситуацій, які необхідно

змодельовати [20].

Таблиця 2.6 – Розраховані ймовірності навантаження математичними методами.

| Фокальний елемент    | $m_1$ | $m_2$ | $m_{DS}$ | $m_{PCR5}$ |
|----------------------|-------|-------|----------|------------|
| Зчитування/запис     | 0.69  | 0.19  | 0.306    | 0.4676     |
| Центральний процесор | 0.34  | 0.1   | 0.069    | 0.1256     |

Згідно з таблиці 2.8 можна зробити висновок, що показники розрахунків математичних методів різняться між собою. Більш правдиві показники показує метод DSmT.

### 2.3.2 Розрахунок оптимізації запитів

Запити сканують великий об'єм даних в пошуку інформації, перевіряючи кожен рядок. Для того, щоб прискорити запит, слід виявити ті стовпці, по яким і необхідно швидко знайти інформацію. Індесування відсортує інформацію в алфавітному порядку. Звернення запиту буде по індексного файлу, який в собі зберігає ідентифікатор та посилання на рядки таблиці. При знайденні хоча б одного значення стовпця, який проіндексований, буде зчитуватися значення індексного файлу та необхідна інформація буде знайдена. Це прискорює пошук, адже не потрібно перевіряти кожен рядок таблиці.

Користуватися індексами потрібно обережно, оскільки при оновленні значень стовпців оновлюються і індекси. Цей процес потребує багато системних ресурсів, несе навантаження на зчитування/запис та на центральний процесор (CPU). Якщо про індексувати помилковий стовбець, продуктивність бази даних буде знижено та будуть марно використовуватися ресурси [23]. В таблиці 2.7 наведені стовпці, які використовуються під час виконання запиту на пошук значень навантажень по курсам.

Кожен розглянутий стовпець потрібно проіндексувати та знайти той стовпець, індексування якого значно впливає на оптимізацію запиту. Важливими показниками для виявлення найбільш оптимізованого варіанту є час виконання запиту та кількість використаних системних ресурсів.

Індексовані стовпці повинні використовуватись в запитах. Для створення індексу дозволяється використовувати будь-який стовпець. Якщо створений індекс не використовується, індекс використовує ресурси без потреби [24]. Описана ситуація негативно впливає на оптимізацію.

Таблиця 2.7 – Стовпці повільного запиту.

| Назва стовпця      | Призначення стовпця  |
|--------------------|--|
| ID_COURSE          | Ідентифікатор навчального предмету.                                  |
| YEAR_PL            | Значення навчального року.   |
| CODE_SPECIALITY    | Код спеціальності групи, в якій навчальний предмет проводиться.      |
| SEMESTR_V          | Значення навчальних семестрів.                                       |
| LEARN_TIME         | Значення часу проведення навчального предмету                        |
| ID_STAFF           | Значення ідентифікатору викладача, який викладає навчальний предмет. |
| ID_SPECIALITY_CATH | Значення категорії спеціальності.                                    |

$$M_{PCRS}(X) = 0.71 * 0.14 + \left[ \frac{0.71^2 * 0.14}{0.71 + 0.14} + \frac{0.14^2 * 0.71}{0.14 + 0.71} \right] = 0.859$$

Згідно з [25] теоретичне навантаження на системні ресурси (cost\_of\_select) методом індексування розраховується за допомогою формул 2.4, 2.5, та 2.6. Розраховуються ресурси кластерного індексування, індексування стовпця та відсутності індексування.

Згідно з формули 2.4 розраховуються необхідні ресурси для індексованої таблиці

$$\text{cost\_of\_select} = HT_I + 1 \quad (2.4)$$

де  $HT_I$  – рівень вкладеності індексу в B-Three дереві.

Значення cost для таблиці з індексами.

$$\text{cost\_of\_select} = 65 + 1 = 66$$

Згідно з формули 2.5 розраховуються необхідні ресурси для кластерного індексу.

$$\text{cost\_of\_select} = HT_I + \left[ \frac{SC(A, R)}{F_R} \right] \quad (2.5)$$

де  $SC(A, R)$  – середнє значення кількості кортежів, які задовольняють умові;  $F_R$  – значення кількості кортежів, які зберігаються в одному блоці даних.

Необхідні ресурси для індексу з кластерами.

$$\text{cost\_of\_select} = 65 + \left[ \frac{42.477}{22.234} \right] = 66.91$$

Згідно з формули 2.6 розраховуються необхідні ресурси для відсутності індексування

$$\text{cost\_of\_select} = HT_I + SC(A, R) \quad (2.6)$$

Ресурси для відсутності індексування:

$$\text{cost\_of\_select} = 65 + 42.477 = 107.477$$

Кожна ймовірність *cost\_of\_select* підраховується під конкретний SQL-запит. Ймовірність для індексованої таблиці становить 27.45%, для кластерного індексу – 27.83%, для таблиці без індексування – 44.71%.

Згідно з [43] розрахунок розміру індексу складається з 7 етапів.

Етап 1. Розрахунок розміру бітової карти NULL (формула 2.7).

Етап 2. Розрахунок розміру даних змінної довжини (формула 2.8)

Етап 3. Розрахунок розміру індексного рядку (формула 2.9)

Етап 4. Розрахунок кількості індексних рядків на сторінку (формула 2.10)

Етап 5. Розрахунок кількості зарезервованих порожніх рядків на сторінці (формула 2.11)

Етап 6. Розрахунок кількості сторінок для зберігання всіх рядків (формула 2.12)

Етап 7. Розрахунок розміру індексу (формула 2.13)

Згідно з формули 2.7, розрахунок бітової карти має вигляд:

$$Leaf\_Null\_Bitmap = 2 + ((Num\_leaf\_cols + 7) / 8) \quad (2.7)$$

де *Num\_leaf\_cols* – кількість індексованих рядків.

Тому розмір бітової карти становить:

$$Leaf\_Null\_Bitmap = 2 + ((68 + 7) / 8) = 11.375$$

Згідно з формули 2.8, розрахунок розміру даних змінної довжини має вигляд:

$$Variable\_leaf\_size = 2 + (Num\_Variable\_leaf\_cols * 2) + Max\_Var\_Leaf\_Size \quad (2.8)$$

де *Num\_variable\_leaf\_cols* – кількість ключових рядків змінної довжини; *Max\_Var\_Leaf\_Size* – розмір в байтах даних рядків змінної довжини.

Тому розмір даних змінної довжини становить:

$$Variable\_Leaf\_Size = 2 + (31 * 2) + 0 = 64$$

Згідно з формули 2.9, розрахунок розміру індексного рядку має вигляд:

$$Leaf\_Row\_Size = Fixed\_Leaf\_Size + Variable\_Leaf\_size + Leaf\_Null\_Bitmap \quad (2.9)$$

де *Fixed\_Leaf\_Size* – фіксований розмір індексу в таблиці в байтах.

Розрахунок розміру індексного рядку становить:

$$Leaf\_Row\_Size = 15 + 64 + 11.375 = 90.375$$

Згідно з формули 2.10 розрахунок кількості індексних рядків на сторінку має вигляд:

$$Leaf\_rows\_per\_page = 8096 / (Leaf\_row\_size + 2) \quad (2.10)$$

де 8096 – значення байтів на сторінку.

Розрахунок кількості індексних рядків на сторінку становить:

$$Leaf\_rows\_per\_page = 8096 / (90.375 + 2) = 87.642$$

Згідно з формули 2.11 розрахунок кількості зарезервованих порожніх рядків на сторінці

має вигляд:

$$\text{Free\_rows\_per\_page} = 8096 * ((100 - \text{Fill\_Factor}) / 100) / (\text{Leaf\_row\_size} + 2) \quad (2.11)$$

де  $\text{Fill\_Factor}$  – значення в БД коефіцієнта заповнюваності порожніх рядків.

Розрахунок кількості зарезервованих порожніх рядків на сторінці становить:

$$\text{Free\_rows\_per\_page} = 8096 * ((100 - 20) / 100) / 90.375 + 2 = 70.114$$

Згідно з формули 2.12 розрахунок кількості сторінок для зберігання всіх рядків має вид:

$$\text{Num\_leaf\_pages} = \text{Num\_row} / (\text{Leaf\_rows\_per\_page} - \text{Free\_rows\_per\_page}) \quad (2.12)$$

де  $\text{Num\_row}$  – кількість рядків в таблиці.

Розрахунок кількості сторінок для зберігання всіх рядків становить:

$$\text{Num\_leaf\_pages} = 3078 / (87.642 - 70.114) = 175.604$$

Згідно з формули 2.13 розрахунок розміру індексу має вид:

$$\text{Leaf\_space\_used} = 8192 * \text{Num\_leaf\_pages} \quad (2.13)$$

Розмір індексу становить:

$$\text{Leaf\_space\_used} = 8192 * 175.604 = 1438547.97 \text{ bytes}$$

Кожна ймовірність  $\text{index\_cost}$  підраховується під конкретний SQL-запит. Ймовірність індексованих рядків в таблиці становить 26.29%, кількість зарезервованих порожніх рядків на сторінці – 21.03%, кількість сторінок для зберігання всіх рядків – 52.67%.

Згідно з формули 2.3 розраховується ймовірність відсутності індексування:

$$M_{PCR5}(X) = 0.2629 * 0.2745 + \left[ \frac{0.2629^2 * 0.2745}{0.2629 + 0.2745} + \frac{0.2745^2 * 0.2629}{0.2629 + 0.2745} \right] = 0.138$$

Згідно з формули 2.3 розраховується ймовірність індексування:

$$M_{PCR5}(X) = 0.2103 * 0.2783 + \left[ \frac{0.2103^2 * 0.2783}{0.2103 + 0.2783} + \frac{0.2783^2 * 0.2103}{0.2103 + 0.2783} \right] = 0.107$$

Показники значень навантажень на зчитування/запис є середнім значенням від кількості значень зчитувань та кількості записів в індексному файлі.

Розбиття основної таблиці на споміжні секції – головне призначення секціонування. За допомогою цього методу не має сенсу сканувати головну таблицю, що покращую продуктивність запити.

Секціонування використовує значно менше системних ресурсів в порівнянні з таблицею без засобів оптимізації.

З використанням формули 2.3 розраховується ймовірність секціонування:

$$M_{PCR5}(X) = 0.5267 * 0.4471 + \left[ \frac{0.5267^2 * 0.4471}{0.5267 + 0.4471} + \frac{0.4471^2 * 0.5267}{0.5267 + 0.4471} \right] = 0.465$$

В таблиці 2.8 порівнюються отримані теоретичні ймовірності показників трьох станів, які були розраховані за допомогою методу DSmT [22] з розрахованими теоретичними показниками.

Таблиця 2.8 – Розраховані ймовірності методом DSmT

| Стан                     | $m_{PCR5}$ |
|--------------------------|------------|
| Відсутність індексування | 0.138      |
| Індексування             | 0.107      |
| Секціонування            | 0.465      |

Обрана найменша ймовірність (0.107) для індексування. Теоретичний розрахунок та злиття даних за допомогою DSmT [22] приблизно однакові.

В таблиці 2.9 наведені використані системні ресурси для обробки запитів з не індексованою таблицею, індексованою та секціонованою.

Таблиця 2.9 – Системні ресурси для трьох станів

| Твердження                        | Cost     |
|-----------------------------------|----------|
| Запит до не індексованої таблиці  | 3413.399 |
| Запит до індексованої таблиці     | 146.348  |
| Запит до таблиці з секціонуванням | 2805.399 |

### 2.3.3 Планування оптимізації запиту

EXPLAIN PLAN використовується в тих випадках, коли ймовірності методів оптимізації мають приблизно однакові значення. EXPLAIN PLAN дає більш детальну інформацію про використання системних ресурсів. Обирається план, який використовує менше системних ресурсів.

EXPLAIN PLAN відображає плани виконання, обрані оптимізатором Oracle для операторів SELECT, UPDATE, INSERT та DELETE. План виконання оператора - це послідовність операцій, які Oracle виконує для запуску оператора.

Дерево рядків є ядром плану виконання. Відображається така інформація [27]:

- Упорядкування таблиць, на які посилається оператор.
- Метод доступу до кожної таблиці, згаданий в операторі.
- Метод з'єднання для таблиць, на які впливають операції приєднання в операторі.

- Операції з даними, такі як фільтр, сортування чи агрегація
- Таблиця плану містить інформацію про наступне [27]:
- Оптимізація, така як вартість та кардинальність кожної операції.
  - Розбиття, наприклад набір доступних розділів.
  - Паралельне виконання, наприклад метод розподілу об'єднаних входів.

В залежності від ключових стовпців, оптимізатор обирає той план, вартість (cost) виконання якого буде найменша. Для аналізу методу потрібно створювати ключові поля та кожного разу слідкувати за вартістю запити.

Без використання жодного ключового стовпця, план виконання запити має вид:

| id | select_type | table  | type | possible_keys | key  | key_len | ref  | rows | Extra  |
|----|-------------|--------|------|---------------|------|---------|------|------|--|
| 1  | SIMPLE      | fint_2 | ALL  | NULL          | NULL | NULL    | NULL | 2777 | Using where; Using temporary; Using filesort |

Рисунок 2.9 - План виконання запити без ключових стовпців

Вартість виконання такого запити становить:

```
| Last_query_cost | 3365.399000 |
```

Рисунок 2.10 - Вартість виконання запити без ключових стовпців

З використання ключового стовпця «ID\_COURSE», план виконання запити має вид:

| id | select_type | table  | type | possible_keys | key  | key_len | ref  | rows |
|----|-------------|--------|------|---------------|------|---------|------|------|
| 1  | SIMPLE      | fint_2 | ALL  | ID_COURSE     | NULL | NULL    | NULL | 2897 |

Рисунок 2.11 - План виконання запити з ключовим стовпцем «ID\_COURSE»

```
| Last_query_cost | 2215.399000 |
```

Рисунок 2.12 - Вартість виконання запити с ключовим полем «ID\_COURSE»

З використання ключового стовпця «SEMESTR\_V», план виконання запити має вид:

| id | select_type | table  | type | possible_keys       | key  | key_len | ref  | rows |
|----|-------------|--------|------|---------------------|------|---------|------|------|
| 1  | SIMPLE      | fint_2 | ALL  | ID_COURSE,SEMESTR_V | NULL | NULL    | NULL | 2709 |

Рисунок 2.13 - План виконання запити з ключовим стовпцем «SEMESTR\_V»

```
| Last_query_cost | 2177.799000 |
```

Рисунок 2.14 - Вартість виконання запити с ключовим полем «SEMESTR\_V»

З використання ключового стовпця «YEAR\_PL», план виконання запити має вид:



```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | type | possible_keys | key | key_len | ref | rows |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | fint_2 | ALL | ID_COURSE,SEMESTR_V,YEAR_PL | NULL | NULL | NULL | 2817 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Рисунок 2.15 - План виконання запиту з ключовим стовпцем «YEAR\_PL»

```

+-----+-----+
| Last_query_cost | 2199.399000 |
+-----+-----+

```

Рисунок 2.16 - Вартість виконання запиту с ключовим полем «YEAR\_PL»

З використання ключового стовпця «CODE\_SPECIALITY», план виконання запиту має вид:

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | type | possible_keys | key | key_len | ref | rows |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | fint_2 | ALL | ID_COURSE,SEMESTR_V,YEAR_PL,CODE_SPECIALITY | NULL | NULL | NULL | 2817 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Рисунок 2.17 - План виконання запиту з ключовим стовпцем «CODE\_SPECIALITY»

```

+-----+-----+
| Last_query_cost | 2199.399000 |
+-----+-----+

```

Рисунок 2.18 - Вартість виконання запиту с ключовим полем «CODE\_SPECIALITY»

В таблиці 2.10 приведені всі вартості плану виконання запитів, серед яких можна зробити висновок про найкращій план.

Таблиця 2.10 – Вартості планів запитів.

| Ключовий стовпець                              | Вартість виконання запиту |
|--|---------------------------|
| Без ключового стовпця                          | 3365.399000               |
| ID_COURSE                                      | 2215.399000               |
| ID_COURSE, SEMESTR_V                           | 2177.799000               |
| ID_COURSE, SEMESTR_V, YEAR_PL                  | 2199.399000               |
| ID_COURSE, SEMESTR_V, YEAR_PL, CODE_SPECIALITY | 2199.399000               |

Проаналізувавши плани виконання запитів були зроблені висновки про методи оптимізації та про кожен їх етап виконання. Було обрано один з методів оптимізації, який використовує меншу кількість системних ресурсів.

Показники EXPLAIN PLAN та розрахунку ймовірностей для індексування мають однакові результати.

Метод індексування під час обробки запитів використовує менше системних ресурсів

ніж секціонування. Індукування зменшує навантаження на зчитування/запис та центральний процесор. Секціонування впливає на дисковий простір бази даних та має гірші показники оптимізації.

| Id | Operation                           | Name      | Cost (%CPU) |
|----|-------------------------------------|-----------|-------------|
| 0  | SELECT STATEMENT                    |           | 1503 (2)    |
| 1  | SORT AGGREGATE                      |           |             |
| 2  | HASH JOIN SEMI                      |           | 1503 (2)    |
| 3  | HASH JOIN                           |           | 915 (0)     |
| 4  | TABLE ACCESS BY INDEX ROWID BATCHED | FIVP_NEW  | 309 (0)     |
| 5  | INDEX RANGE SCAN                    | ID_COURSE | 295 (0)     |

PLAN\_TABLE\_OUTPUT

|   |                   |          |         |
|---|-------------------|----------|---------|
| 6 | TABLE ACCESS FULL | CSVS_NEW | 693 (0) |
| 7 | TABLE ACCESS FULL | CSVS_NEW | 693 (0) |

Рисунок 2.19 – Використання ресурсів методом індексування.

| Id | Operation                | Name     | Rows | Bytes | Cost (%CPU) |
|----|--------------------------|----------|------|-------|-------------|
| 0  | SELECT STATEMENT         |          | 1    | 34    | 1873 (3)    |
| 1  | SORT AGGREGATE           |          | 1    | 34    |             |
| 2  | HASH JOIN SEMI           |          | 730  | 24820 | 1873 (3)    |
| 3  | TABLE ACCESS FULL        | CSVS_NEW | 968  | 15488 | 681 (0)     |
| 4  | PARTITION RANGE ITERATOR |          | 1076 | 19368 | 1208 (7)    |
| 5  | TABLE ACCESS FULL        | F_PART   | 1076 | 19368 | 1208 (7)    |

Рисунок 2.20 – Використання ресурсів методом секціонування

В результаті розрахунків методів оптимізації було отримано три ймовірності. Для прийняття рішення щодо оптимізації запитів необхідно слідкувати за показниками обраних метрик (Disk I/O, CPU) та якщо їх значення перевищують порогові значення та розрахунковий час обробки запиту. В разі перевищенні порогових значенні необхідно сповіщувати користувача про пікове навантаження системи. Для оптимізації необхідно застосовувати метод індексування, який має найбільшу ймовірність оптимізації, має найбільший відсоток корисності та використовує найменше значення системних ресурсів серед інших методів.

Загальний алгоритм складається з аналізу метрик баз даних, вибору декількох основних з них, аналізу та злиття даних по метрикам. Після злиття даних метрик розглядаються методи оптимізації запитів та аналіз їх результатів. Розрахунок їх ймовірностей та прийняття рішень щодо оптимізації запитів.

Блок-схема алгоритму виявлення та оптимізації повільного запиту в додатку А.

## Висновки до другого розділу

Стан системи під час обробки запитів може бути непередбачуваний. Використання великої кількості системних ресурсів, неправильна структура запиту та схема бази даних – всі ці фактори впливають на продуктивність бази даних. Виявити проблему продуктивності можна з використанням обраних метрик. Якщо значення метрик не мають критичних значень, проблема оптимізації не в структурі бази даних. Тобто не використовується максимальна продуктивність системи. Велика кількість зчитувань даних впливає на простоті процесору та навпаки, під час обробки даних процесору формується черга на виконання. За допомогою журналу повільних запитів можна виявити повільні запити та з'ясувати час виконання цих запитів, кількість сканованих стовбців під час обробки запиту, загальну кількість блокувань. Щоб кожного разу не слідкувати за показниками метрик, було здійснено злиття даних метрик за допомогою підрахованих ймовірностей за допомогою ВВА (basic belief assignment). За отриманим значенням злиття було зроблено висновок, що систему можна оптимізувати методами оптимізації. Була розрахована ймовірність оптимізації кожного методу та виявлено, що метод індексування має найбільшу ймовірність для оптимізації запиту.

Якщо метрики не сигналізують про навантаження системи, а запит виконується повільно, треба шукати проблеми в самому запиті. Складну структуру запиту можна розділити на прості під запити та таким чином слідкувати за показниками системи.

Щоб не перевіряти метрики, були розраховані приблизні ймовірності для розглянутих методів оптимізації. Серед розглянутих методів оптимізації найкращі показники має індексування. Індексування прискорює час обробки запитів та використовує менше всіх системних ресурсів.

## Література до другого розділу

1. Oracle All Things SQL. How to Create and Use Indexes in Oracle Database - [Електронний ресурс] / Oracle Corporation - Режим доступу: [www. URL: https://blogs.oracle.com/sql/how-to-create-and-use-indexes-in-oracle-database/](http://www.oracle.com/technetwork/blogs/sql/how-to-create-and-use-indexes-in-oracle-database/) - August 10, 2017.
2. DZone. Database Zone. Top Metrics to Monitor in Your MySQL Databases (Part 1) - [Електронний ресурс] / DZone's Technical Library - Режим доступу: [www. URL: https://dzone.com/articles/top-performance-metrics-to-monitor-in-your-mysql-d-1](http://www.dzone.com/articles/top-performance-metrics-to-monitor-in-your-mysql-d-1) - 27.10.2017 р.
3. Oracle® Database. Quality of Service Management User's Guide. 12c Release 1 (12.1) E49047-07 - [Електронний ресурс] / Oracle Corporation - Режим доступу: [www. URL: https://docs.oracle.com/database/121/APQOS/E49047-07.pdf](http://docs.oracle.com/database/121/APQOS/E49047-07.pdf) / 42-43 с. - January 2017.
4. ORACLE Help Center. Enterprise Manager System Monitoring Plug-in for Oracle Times Ten In-Memory Database User's Guide. Viewing Metrics - [Електронний ресурс] / Oracle Corporation - Режим доступу: [www. URL: https://docs.oracle.com/cd/E24628\\_01/timesten.121/e28645/metrics.htm#TTEMP132/](http://www.oracle.com/technetwork/docs/enterprise-manager/enterprise-manager-system-monitoring-plug-in-for-oracle-times-ten-in-memory-database-user-s-guide-viewing-metrics-2888871.pdf) - 15.08.2015р.
5. ORACLE Help Center. Enterprise Manager System Monitoring Plug-in Metric Reference Manual for Non Oracle Database Management. Lock Waits Metrics - [Електронний ресурс] / Oracle Corporation - Режим доступу: [www. URL: https://docs.oracle.com/cd/E73210\\_01/EMMDA/GUID-58A6EA5A-FB66-45AF-AA57-68BC0A0254ED.htm#EMMDA391/](http://www.oracle.com/technetwork/docs/enterprise-manager/enterprise-manager-system-monitoring-plug-in-metric-reference-manual-for-non-oracle-database-management-lock-waits-metrics-2888871.pdf) - 31.10.2017р.
6. Database Monitoring Metrics - [Електронний ресурс] / AppDynamics APM Platform Documentation - Режим доступу: [www. URL: https://docs.appdynamics.com/display/PRO45/Database+Monitoring+Metrics/](http://www.appdynamics.com/docs/PRO45/Database+Monitoring+Metrics/) - 25.04.2019р.
7. Yong Guo, Marcin Biczak, Ana Lucia Varbanescu та ін. How Well do Graph-Processing Platforms Perform? An Empirical Performance Evaluation and Analysis / 2014 IEEE 28th International Parallel & Distributed Processing Symposium – 2014р.
8. Janet L. Wiener, Stavros Harizopoulos, Mehul A. Shah, Goetz Graefe. Database join optimized for flash storage / Hewlett Packard Enterprise Development LP. United States Patent – 3 листопада 2015 р.
9. Новиков Б. А. Основы технологий баз данных: учеб. пособие / Б. А. Новиков, Е. А. Горшкова; под ред. Е. В. Рогова. — М.: ДМК Пресс, 2019 - 21-22 с.
10. SHOW PROCESSLIST Syntax - [Електронний ресурс] / MySQL 8.0 Reference Manual - Режим доступу: [www. URL: https://dev.mysql.com/doc/refman/8.0/en/show-processlist.html/](http://www.mysql.com/doc/refman/8.0/en/show-processlist.html) - 12.09.2016р.
11. The General Query Log - [Електронний ресурс] / MySQL 8.0 Reference Manual - Режим доступу: [www. URL: https://dev.mysql.com/doc/refman/8.0/en/query-log.html/](http://www.mysql.com/doc/refman/8.0/en/query-log.html) - 17.12.2018р.
12. Showing running queries in MySQL - [Електронний ресурс] / the electronic toolbox -

Режим доступу: www. URL: <https://www.electrictoolbox.com/show-running-queries-mysql/> - 18.02.2019р.

13. ИТ Блог. Администрирование серверов на основе Linux (Ubuntu, Debian, CentOS, openSUSE) - [Электронный ресурс] / AndreyEx - Режим доступу: www. URL: <https://andreyex.ru/bazy-dannyx/baza-dannyx-mysql/pokazat-protsess-v-mysql/> - 03.06.2019р.

14. MySQL. General Query Log - [Электронный ресурс] / RIP Tutorial - Режим доступу: www. URL: <https://riptutorial.com/mysql/example/18052/general-query-log/> - 06.08.2016р.

15. Disk I/O % Utilization - [Электронный ресурс] / Mongo DB Documentation - Режим доступу: www. URL: <https://docs.atlas.mongodb.com/reference/alert-resolutions/disk-io-utilization/> - 25.01.2019р.

16. Скорость загрузки сайта имеет значение - [Электронный ресурс] / Справка Google - Режим доступу: www. URL: <https://support.google.com/google-ads/answer/9069088?hl=ru/> - 11.07.2018р.

17. Pekka Enberg, Ashwin Rao, Sasu Tarkoma. I/O Is Faster Than the CPU – Let’s Partition Resources and Eliminate (Most) OS Abstractions / University of Helsinki – 1-2 с. - 13.05.2019р.

18. Fuyuan Xiao. Generalization of Dempster–Shafer theory: A complex belief function - [Электронный ресурс] / arXiv:1906.11409v1 [cs.AI] - Режим доступу: www. URL: <https://arxiv.org/pdf/1906.11409.pdf> - 27 Jun 2019.

19. Dezert-Smarandache Theory (DSmT) of Plausible. What is DSmT ? - [Электронный ресурс] / University of New Mexico - Режим доступу: www. URL: <http://fs.unm.edu/DSmT.htm/> - 18 January 2019.

20. Audun Jøsang, Simon Pope. Dempster’s Rule as Seen by Little Coloured Balls / Computational Intelligence, Volume 00, Number 000, 2012 - January 2010.

21. Веригіна І. В., Островська О. В. Теорія ймовірностей та математична статистика. Частина 1. Випадкові події. - [Электронный ресурс] / Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» - Режим доступу: www. URL: [http://ela.kpi.ua/bitstream/123456789/23501/1/NP%28T\\_Ym%29\\_1.pdf](http://ela.kpi.ua/bitstream/123456789/23501/1/NP%28T_Ym%29_1.pdf) – 15-16, (2.1) - 24.05.2018 р.

22. Jean Dezert, Florentin Smarandache. Non Bayesian Conditioning and Deconditioning / Advances and Applications of DSmT for Information Fusion. Collected Works. Volume 4 – 11-12с., - 2015р.

23. Darl Kuhn, Sam R.Alapati and Bill Patfield. Expert Indexing in Oracle Database 11g. Maximum Performance for your Database / The experts voice in Oracle - Apress - 1-2с., 2009

24. Oracle® Database Application Developer's Guide – Fundamentals 10g Release 1 (10.1) - [Электронный ресурс] / Oracle Corporation - Режим доступу: www. URL: [https://docs.oracle.com/cd/B12037\\_01/appdev.101/b10795/adfn\\_in.htm/](https://docs.oracle.com/cd/B12037_01/appdev.101/b10795/adfn_in.htm/) - 12.05.2019р.

25. UC Davis. Query Processing and Optimization - [Электронный ресурс] / Dept. of Computer Science - Режим доступа: www. URL: <https://web.cs.ucdavis.edu/~green/courses/ecs165a-w11/8-query.pdf> - 139-143с. - 01.05.2018

26. Оценка размера некластеризованного индекса - [Электронный ресурс] / Документация по SQL - Режим доступа: www. URL: <https://docs.microsoft.com/ru-ru/sql/relational-databases/databases/estimate-the-size-of-a-nonclustered-index?view=sql-server-ver15> – 13.03.2017

27. MySQL Explain Example. Explaining MySQL EXPLAIN using StackOverflow data - [Электронный ресурс] / EverSQL - Режим доступа: www. URL: <https://www.eversql.com/mysql-explain-example-explaining-mysql-explain-using-stackoverflow-data/> October 1, 2018.

## РОЗДІЛ 3

### РОЗРОБКА МЕТОДИКИ ФІКСУВАННЯ ПОВІЛЬНИХ ЗАПИТІВ

Під час обробки запиту системні метрики не повинні перевищувати раніше встановлені порогові значення. Якщо показники перевищили порогові значення, необхідно аналізувати запит. Розроблена методика сповіщає користувача про відхилення значень порогових значень та дає змогу ознайомитись з показниками змінних стану базу даних. Оптимізовані запити не мають перевищувати показники метрик.

#### 3.1 Проектування веб-інтерфейсу методики

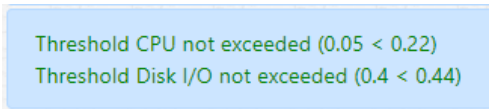
Важливо проектувати простий та зрозумілий дизайн для методики. Користувачам необхідно відразу зрозуміти те, що методика від нього вимагає. Користувачи не читають вміст сторінки, вони його аналізують [1]. Тому інтерфейс методики не має містити нічого зайвого. Методика повинна бути максимально простою та приємною в користуванні [2]. Методика мусить відповідати вимогам того, для чого вона буда створена та вирішувати поставлені перед нею задачі. Необхідно заздалегідь розробити шаблон інтерфейсу та виявити майбутні недоліки з його використанням. Розміщення об'єктів має бути гармонійним та водночас компактним [3]. Під час проектування дизайну необхідно розглянути багато факторів, які можуть зацікавити користувачів, щоб ті звернули увагу на те, що їм пропонує веб додаток [4]. Гарний дизайн та навігація робить додаток простим в користуванні. Дизайн при такому поєднанні не відіграє великої ролі завдяки тому, що відразу зрозуміла мета додатку [5].

Створення інтерфейсу – дуже відповідальна справа. Необхідно радитись не з одною людиною та аналізувати критику. Тестувати інтерфейс повинна окрема людина, розробник може не знайти в інтерфейсі помилок [6].

Необхідно планувати стратегії, щоб інтерфейс став візитною картою розробленого додатку [7].

#### 3.2 Фіксування повільних запитів розробленою методикою

Розроблена методика фіксує кількість запитів та відображає їх. Коли навантаження на систему перевищує встановлені порогові значення для метрик, методика спрацьовує та відображає повідомлення.



```
Threshold CPU not exceeded (0.05 < 0.22)
Threshold Disk I/O not exceeded (0.4 < 0.44)
```

Рисунок 3.1 – Фіксування методикою значень системи

В процесі обробки запиту методика відображає користувачу поточний стан системи, який відображає:

- Uptime. Час з моменту запуску сервера.
- Threads. Кількість потоків, які обробляють запит.
- Questions. Кількість операторів оброблених сервером.
- Slow queries. Кількість повільних запитів.
- Open tables. Кількість таблиць з яких відбувається зчитування інформації.
- Queries per second avg. Середня кількість оброблених запитів в секунду.

**Server status:**  
 Threads: 3  
 Slow queries: 82  
 Flush tables: 1  
 Open tables: 5  
 Queries per second avg: 0.985

Рисунок 3.2 – Відображення поточного стану системи

Методика сканує показники використаних системних ресурсів та відображає навантаження на систему в виді графіків. Частота оновлення графіків складає 5 сек. Графічне відображення стану дуже зручне, показується фактичний стан системи під час обробки запиту. Графіки відображають навантаження на центральний процесор та на кількість зчитувань/записів. Раніше було встановлено, що саме ці дві метрики є основними для перевірки стану системи. Згідно з графіками було встановлено, що показники центрального процесору в багатьох випадках перевищують порогове значення. Показники метрики зчитування/запису завжди нижче встановленого порогового значення.

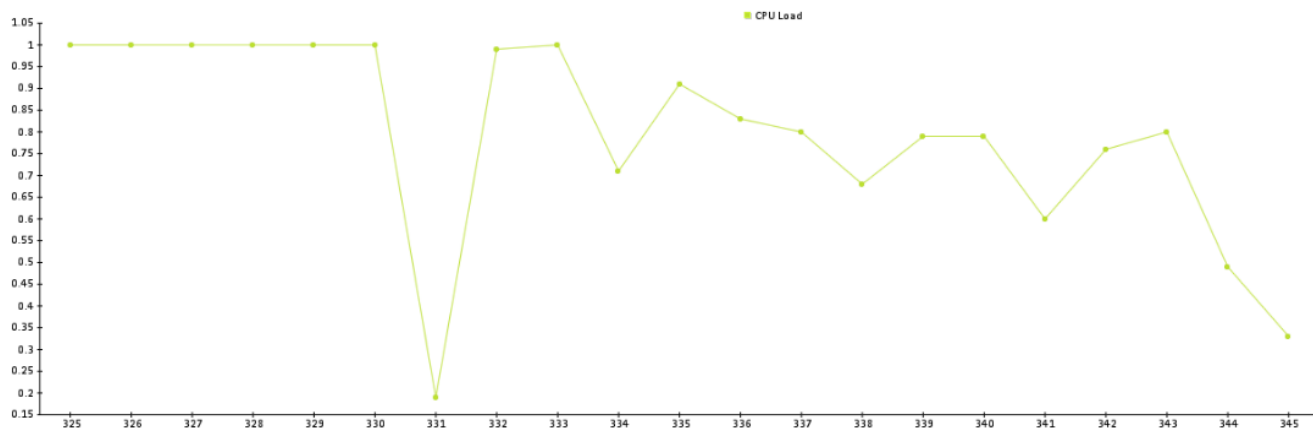


Рисунок 3.3 – Графік навантаження на центральний процесор с веб інтерфейсу методики

Методика зафіксувала велике навантаження на центральний процесор та згенерувала графік на основі даних про навантаження. Максимальне зафіксоване значення навантаження



становить 1, мінімальне – 0.2.

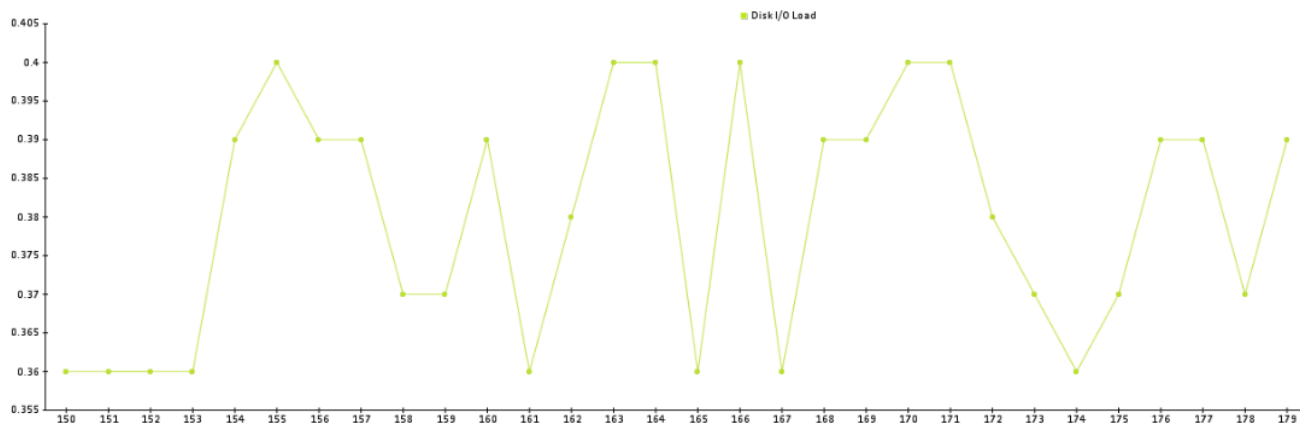


Рисунок 3.4 – Графік навантаження на зчитування/запис с веб інтерфейсу методики

Методика не виявила перевищення порогового значення для зчитування/запису. Навантаження не перевищує значення 0.4. Мінімальне значення навантаження становить 0.36.

Методика передбачає аналіз трьох станів запитів:

- Без оптимізації.
- З ви користуванням індексування.
- З використанням секціонування.

Методика передбачає очищення графіків під час обробки запитів. Це необхідно при фіксуванні повільного запиту та спостереженню впливу цього запиту на показники системи.

Графіки формуються за допомогою бібліотеки рChart [8] з використанням мови програмування PHP [9]. Дизайн методики сформований за допомогою таблиці каскадних CSS [10] та фреймворку Bootstrap [11]. Взаємодія з елементами сторінки методики за допомогою JavaScript [12] та надсилання фонових запитів на сервер за допомогою AJAX [13]. Використовується база даних MySQL [14]. Використовується сервер Apache [15].



Рисунок 3.5 – Інтерфейс методики

### 3.3 Аналіз не оптимізованого стану запитів

Аналізуються запити, які не мають оптимізації. Головними критеріями аналізу є час обробки, кількість перевищення порогових значень та середнє навантаження під час обробки запитів.

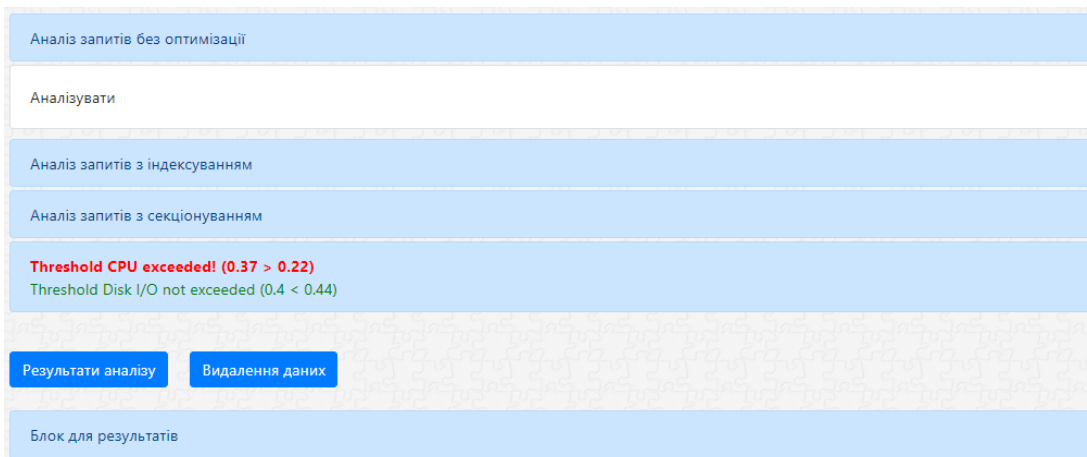


Рисунок 3.6 – Інтерфейс аналізу не оптимізованих запитів

Після запуску аналізу формуються графіки навантаження на зчитування/запис та навантаження на центральний процесор. Період оновлення запитів становить 5 сек. На графіках фіксуються перевищення порогових запитів та розраховується загальна ймовірність навантаження на систему. Розрахунок відбувається на основі отриманих значень метрик.

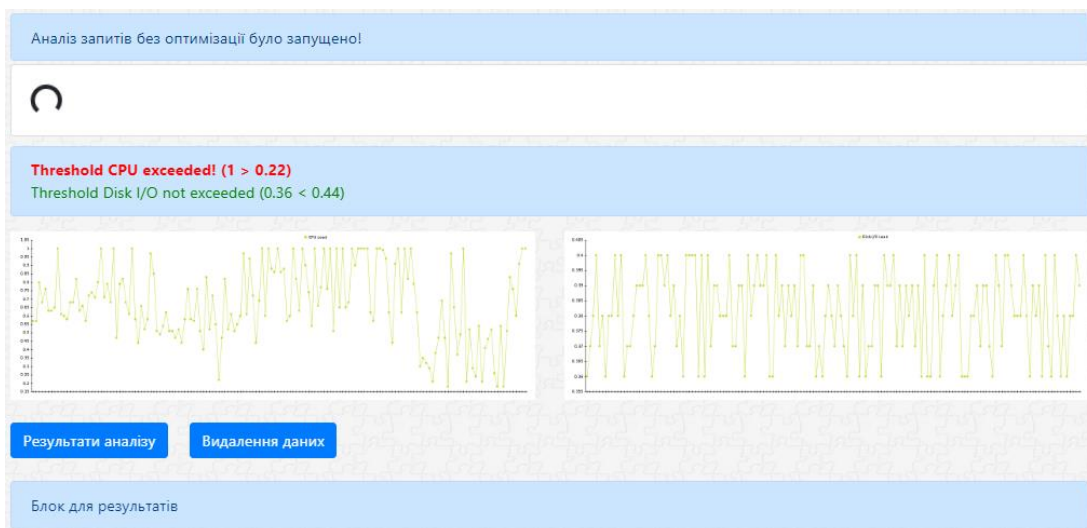


Рисунок 3.7 – Графіки навантаження на систему не оптимізованих запитів

Навантаження на центральних процесор в багатьох випадках перевищують порогові значення. Показники навантажень перебувають в діапазоні від 25% до 97%. Значенням навантаження буде вважатись середнє значення отриманих показників.

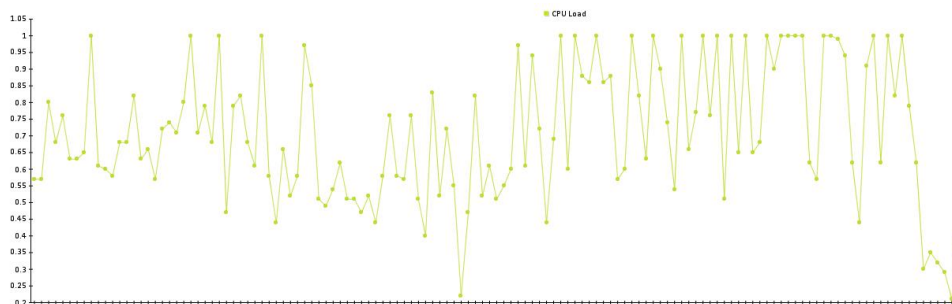


Рисунок 3.8 – Навантаження на центральний процесор

Показники зчитування/запису не перевищують порогові значення. Значення навантаження завжди менше встановленого пікового значення.

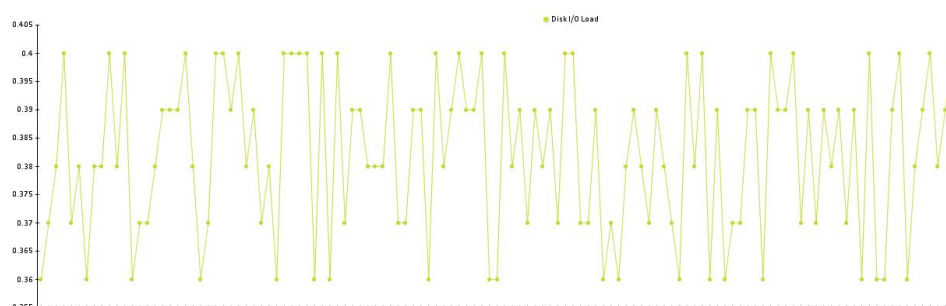


Рисунок 3.9 – Навантаження на зчитування/запис

Навантаження на зчитування/запис становить від 36% до 44%. Це свідчить про те, що не оптимізовані запити навантажують центральний процесор. Необхідна оптимізація, яка зменшить навантаження на процесор та зменшить кількість зчитування/запису.

### 3.4 Аналіз запитів з методом індексування

Для усунення проблем з навантаженням на систему, використовуються методи оптимізації. Для оптимізації запиту з методом індексування необхідно обрати пункт аналізу в веб інтерфейсі методики.

Рисунок 3.10 – Активація аналізу запиту з методом індексування

Аналізується час обробки запиту, кількість перевищення порогових запитів та середнє навантаження на систему.

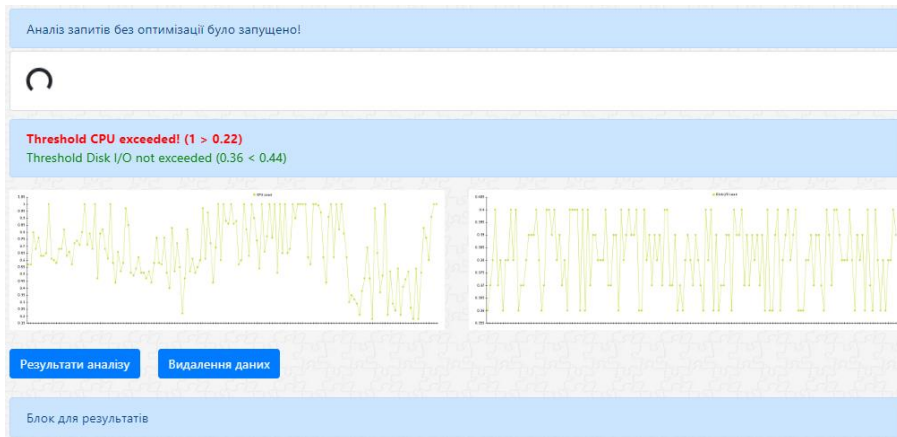


Рисунок 3.11 – Початок аналізу

З початком аналізу інші стани аналізу стають не активними, поки не закінчиться аналіз активованого стану. Одночасно аналіз всіх станів в методиці неможливий, це приведе до неточності отриманих даних та сформує хибний висновок після аналізу всіх станів.

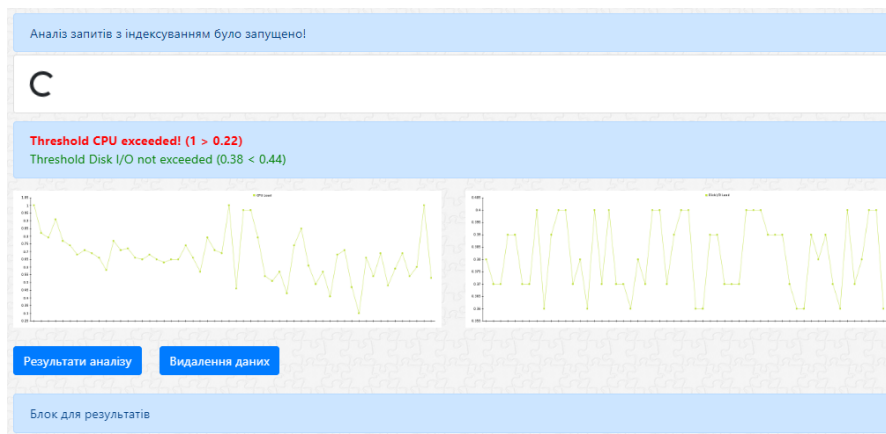


Рисунок 3.12 – Формування графіків під час обробки запиту

Навантаження на центральний процесор зменшилось в порівнянні з не оптимізованим станом.

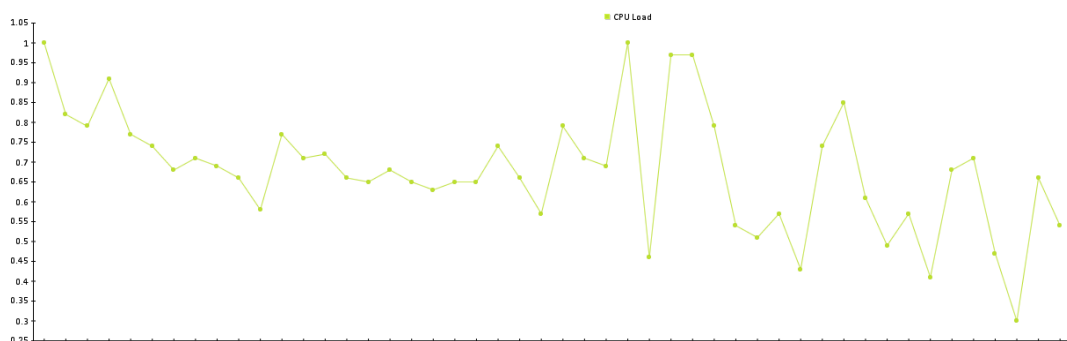


Рисунок 3.13 – Графік навантаження на центральний процесор

Навантаження складає від 30% до 100%. Максимальне пікове навантаження перевищується менше разів. Середнє значення навантаження під час обробки таких запитів становить 62%.

Навантаження на зчитування/запис не перевищуються.

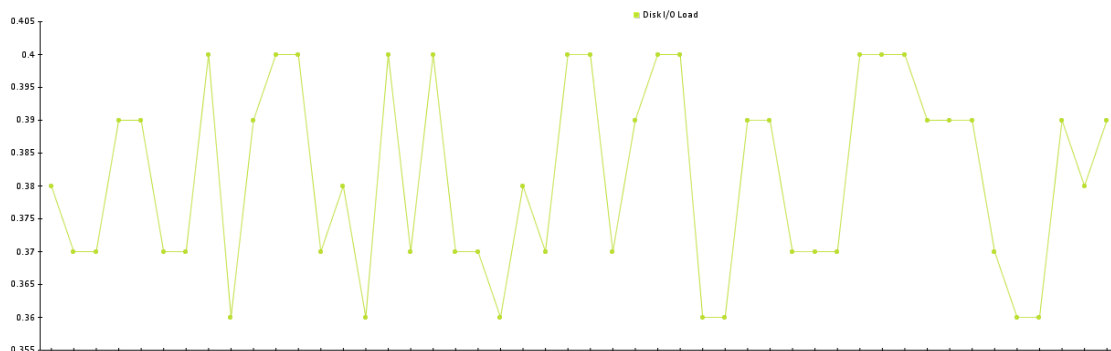


Рисунок 3.14 – Навантаження на зчитування/запис

Метод індексування зменшив навантаження на центральний процесор. Зчитування/запис не перевищує встановлені порогові значення.

### 3.5 Аналіз запитів з методом секціонування

Перевіряються запити з використанням методом секціонування, який розбиває таблицю на окремі таблиці. Оцінюється час обробки запитів, кількість перевищень порогових значень та середнє навантаження на систему. Після аналізу всіх станів запитів, методика робить висновок про найкращі показники.

Під час аналізу запиту був сформований графік, який фіксує навантаження на центральних процесор. Під час обробки запиту було 92 рази перевищено порогове значення, при цьому середнє навантаження на процесор складає 64%.

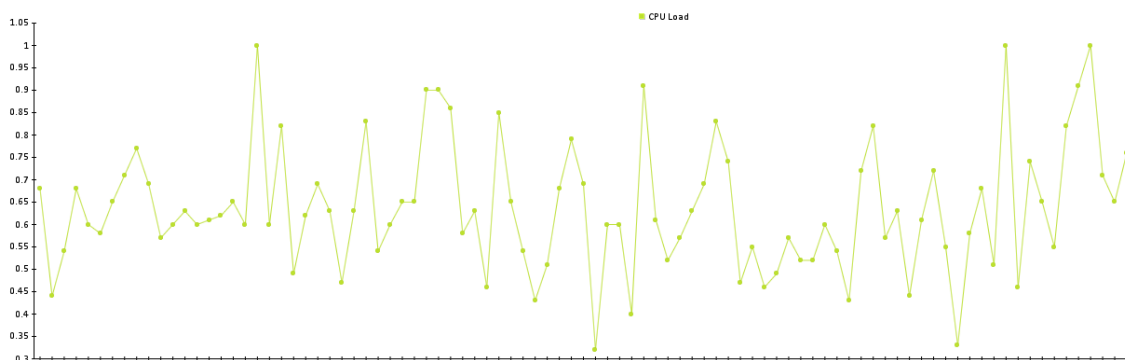


Рисунок 3.15 – Графік навантаження на центральний процесор

Значення зчитування/запису не перевищує порогові значення. Це свідчить про те, що кількість зчитувань в повільному запиті не має критичних показників для системи.

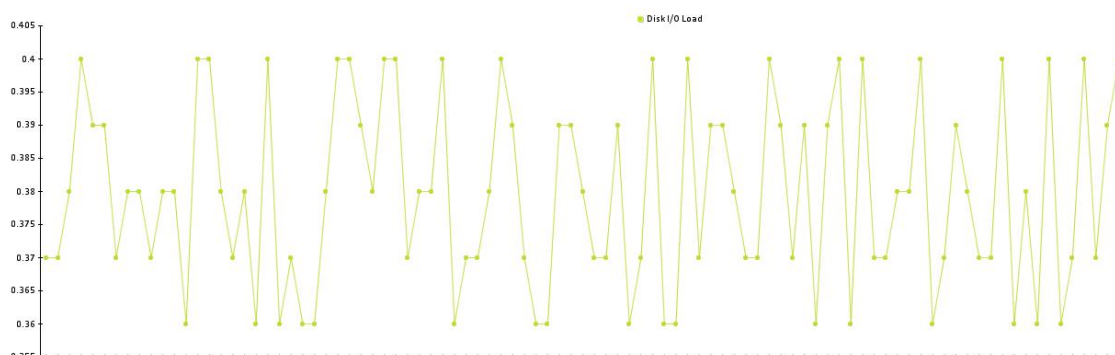


Рисунок 3.16 – Графік навантаження на зчитування/запис

Навантаження належить діапазону від 36% до 44%. Проблему оптимізації необхідно шукати в аналізі інших метрик. В даному випадку показники навантаження на центральний процесор мають велике навантаження та значно перевищує встановлені норми. Головна проблема в тому, що така ситуація сповільняє роботу базу даних та очікування результатів обробки значно збільшується. З урахуванням великої кількості користувачів, час очікування буде збільшуватись та є ймовірність втрати даних.

Під час аналізу всіх станів було встановлено, що показники центрального процесору мають навантаження, які з часом перевищують порогові значення. Метою методики є знаходження того стану запитів, який обробляється швидше, менше всього перевищую порогові значення на має менше навантаження на систему.

Висновком аналізу всіх станів є таблиця з показниками. Показники перетворюються в ймовірність на обирається найменша з них. Розрахована ймовірність – показник навантаження системи. Обирається найменша ймовірність навантаження системи. Методика, на основі отриманих показників, рекомендує метод оптимізації.

| Всі стани запитів              |                 |          |              |          |               |          |
|--------------------------------|-----------------|----------|--------------|----------|---------------|----------|
|                                | Без оптимізації |          | Індексування |          | Секціонування |          |
|                                | CPU             | Disk I/O | CPU          | Disk I/O | CPU           | Disk I/O |
| Перевищення порогу, кіл-ть     | 149             | 0        | 56           | 0        | 92            | 0        |
| Середнє навантаження           | 0.67            | 0.381    | 0.673        | 0.381    | 0.64          | 0.38     |
| <b>Завантаженість системи:</b> | 0.399           |          | 0.156        |          | 0.242         |          |

**Індексовані** запити найбільш ефективні. Вони мінімально перевищують порогові значення та мають менше навантаження на систему

Рисунок 3.17 – Висновок методики після аналізу всіх станів

Час обробки запитів має найменші показники у індексування. Метод індексування менше всіх інших станів перевищував порогові значення. Методика пропонує використовувати метод індексування.

### 3.6 Використання системних ресурсів

Необхідно фіксувати показники використання системних ресурсів бази даних до та під час запущеного повільного запиту. Показники системи за певний проміжок часу з розрахунком ймовірності навантаження. Порівнювати отримані дані з встановленими пороговими значеннями для кожного з системних ресурсів. Графіки навантаження системи відображають її стани в певні проміжки часу та на їх основі приймаються рішення про оптимізацію запитів. Якщо значення системних ресурсів не перевищують встановлені значення, запит не потрібно оптимізувати. Якщо час обробки запиту перевищує встановлені норми, а використання системних ресурсів при цьому мінімальний, необхідно змінювати структуру самого запиту.

Стан системи до запущеного повільного запиту.

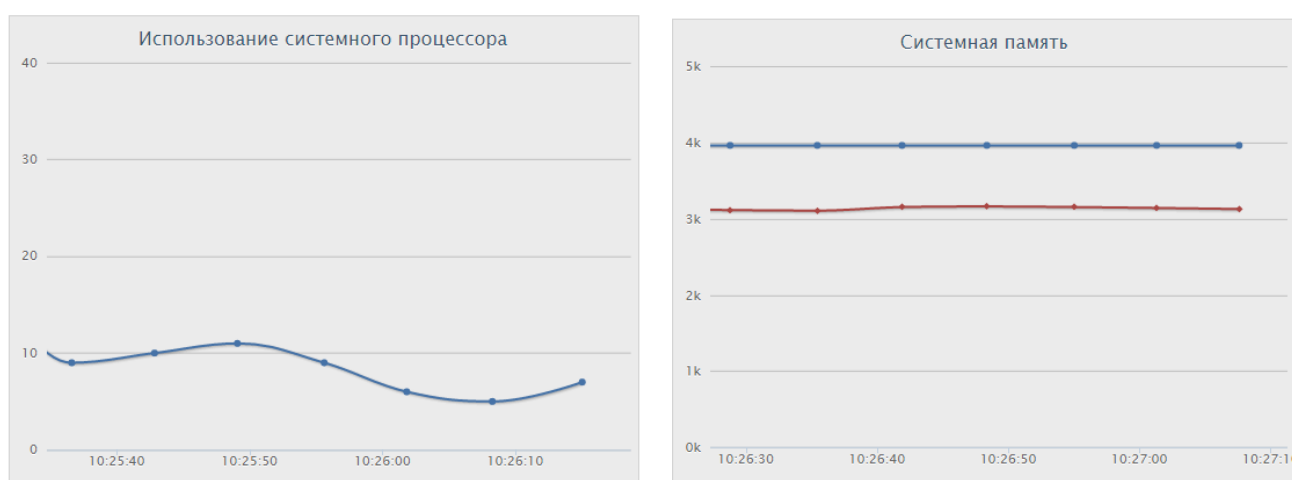


Рисунок 3.18 – Показники системних ресурсів

Проміжок часу моніторингу використання системних ресурсів становить 30 сек.

Для центрального процесору зафіксовані показник навантаження від 5 до 11%. Середнє навантаження процесору становить 8.37%.

Системна пам'ять (ОЗУ) навантажена від 234 МБ до 289 МБ. Середнє значення навантаження на ОЗУ становить 239.1 МБ.

Після запуску повільних запитів показники навантаження системи відрізняються. Навантаження збільшується, про що фіксується на наступних графіках.

Навантаження на центральний процесор становить від 8% до 64%. Середнє навантаження на процесор становить 32.5%.

Системна пам'ять (ОЗУ) навантажена від 290 МБ до 411 МБ. Середнє значення навантаження на ОЗУ становить 350.1 МБ.

Для запуску оптимізатору запиту необхідно, щоб навантаження на ЦП перевищувало відмітку 44% (0.44 ймовірність). За 30 с. було зафіксовано 6 станів, коли навантаження на ЦП перевищувало встановлений ліміт. В цьому випадку необхідне втручання оптимізатору запиту.

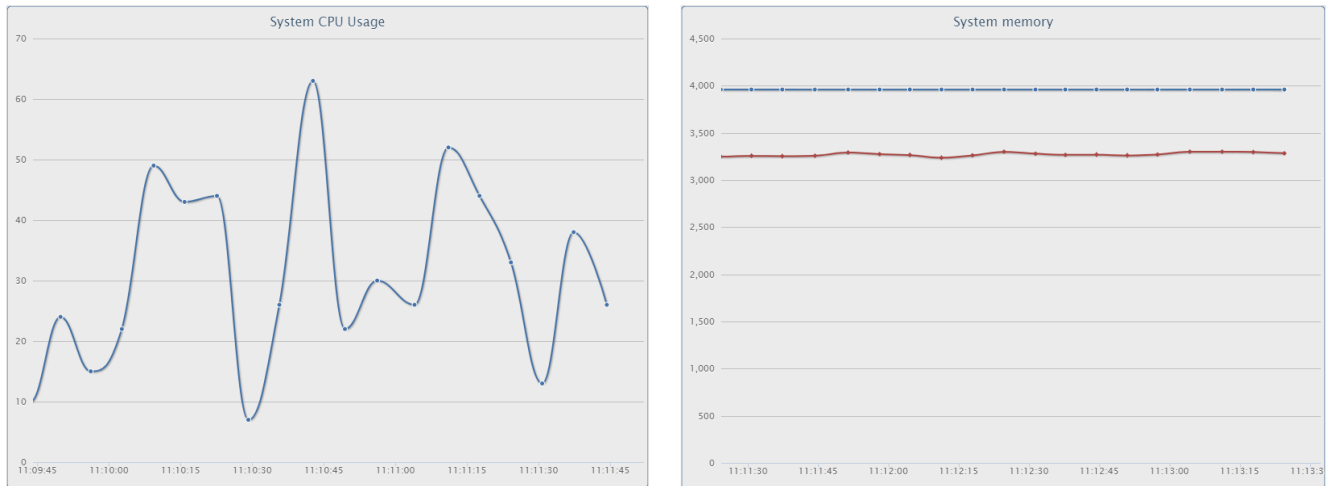


Рисунок 3.19 – Показники системних ресурсів

В таблиці 3.6 порівняння використання системних ресурсів до та під час обробки повільного запиту.

Таблиця 3.6 – Порівняння використання системних ресурсів

| Стан                              | ЦП, % | ОЗУ, МБ |
|-----------------------------------|-------|---------|
| До запуску повільних запитів      | 8.37  | 239.1   |
| Під час запуску повільних запитів | 32.5  | 350.1   |

Згідно з отриманих результатів можна зробити висновок, що навантаження на центральний процесор збільшилось в 3.88 разів, а на ОЗУ в 1.46.

Аналіз реальних системних показників показав, що навантаження на центральний процесор більші, ніж на всі інші системні метрики. При цьому навантаження збільшується під час запуску запитів. Показники методики та реальних даних мають розбіжності. Зафіксовані різні проміжки часу. На реальних графіках зображені стани, коли показники метрик не перевищують встановлені порогові значення.



## Висновки до третього розділу

Для фіксування повільних запитів та сповіщення користувача про перевищення порогових значень метрик буда створена методика. Розроблена методика відображає стан системи під час обробки запиту. Вона фіксує:

- Кількість повільних запитів.
- Кількість таблиць, з яких відбувається зчитування.
- Середню кількість оброблених запитів в секунду.
- Кількість потоків, які обробляють запити.

Методика аналізує:

- Показники не оптимізованих запитів.
- Показники запитів з використанням методу індексування.
- Показники запитів з використанням секціонування.

Методика сповіщає:

- Про перевищення пікових значень метрик.
- Про час обробки запитів.

Методика допомагає фіксувати повільні записи, відстежувати стан системи та аналізувати порогові значення для обраних метрик.

Після аналізу всіх станів запитів, методика аналізує показники та надає рекомендації щодо найкращого методу оптимізації.

## Література до третього розділу

1. Steve Krug. Don't Make Me Think: A Common Sense Approach to Web Usability / Steve Krug. - New Riders Publishing, 2009 – 21-23с.
2. Jennifer Niederst Robbins. Learning Web Design / Jennifer Niederst Robbins – O'Reilly Media, Inc, 2012 – 5-6с.
3. Jason Beaird. The Principles of Beautiful Web Design / Jason Beaird - SitePoint Pty. Ltd, 2010 – 2-3с.
4. Jenifer Tidwell. Designing interfaces / Jenifer Tidwell - O'Reilly Media, Inc, 2011 - 3-4с.
5. Theresa Neil. Mobile Design Pattern Gallery: UI Patterns for Mobile Applications / Theresa Neil - O'Reilly Media, Inc, 2012, 2-3с.
6. Jeff Gothelf. Lean UX: Designing Great Products with Agile Teams / Jeff Gothelf - O'Reilly Media, Inc, 2016, 6-7с.
7. Jaime Levy. UX Strategy: How to Devise Innovative Digital Products that People Want / Jaime Levy - O'Reilly Media, Inc, 2015, 6-8с.
8. pChart Official Documentation - [Електронний ресурс] / pChart 2.0 - Режим доступу: [www. URL: http://wiki.pchart.net/documentation/](http://wiki.pchart.net/documentation/) 2010
9. PHP 7.3.1 - [Електронний ресурс] / PHP Documentation - Режим доступу: <https://www.php.net/> 18.12.2019
10. Cascading Style Sheets – [Електронний ресурс] / World Wide Web Consortium - Режим доступу: <https://www.w3.org/Style/CSS/Overview.en.html> / 2017
11. Bootstrap - [Електронний ресурс] / Twitter Bootstrap Documentation - Режим доступу: [https://bootstrap-4.ru/docs/4.3.1 /](https://bootstrap-4.ru/docs/4.3.1/) 03.10.2017
12. JavaScript - [Електронний ресурс] / JavaScript Documentation - Режим доступу: <https://www.javascript.com/>
13. Введение в Ajax - [Електронний ресурс] / Site Javascript.ru - Режим доступу: <https://learn.javascript.ru/ajax-intro>
14. MySQL - [Електронний ресурс] / Oracle Inc. - Режим доступу: <https://dev.mysql.com/doc/>
15. Apache HTTP SERVER - [Електронний ресурс] / Apache Software Foundation - Режим доступу: <https://httpd.apache.org/>

## РОЗДІЛ 4

### ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

#### 4.1 Загальні питання з охорони праці

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» [1] визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

При роботі з обчислювальною технікою змінюються фізичні і хімічні фактори навколишнього середовища: виникає статична електрика, електромагнітне випромінювання, змінюється температура і вологість, рівень вміст кисню і озону в повітрі. Повітря забруднюється шкідливими хімічними речовинами антропогенного походження за рахунок деструкції полімерних матеріалів, які використовуються для обробки приміщень та обладнання. Неправильна організація робочого місця сприяє загальному і локальній напрузі м'язів шії, тулуба, верхніх кінцівок, викривлення хребта і розвитку остеохондрозу. На всіх підприємствах, в установах, організаціях повинні створюватися безпечні і нешкідливі умови праці. Забезпечення цих умов покладається на власника або уповноважений ним орган (далі роботодавець). Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. Роботодавець повинен впроваджувати сучасні засоби техніки безпеки, які запобігають виробничому травматизмові, і забезпечувати санітарно-гігієнічні умови, що запобігають виникненню професійних захворювань працівників. Він не має права вимагати від працівника виконання роботи, поєднаної з явною небезпекою для життя, а також в умовах, що не відповідають законодавству про охорону праці. Працівник має право відмовитися від дорученої роботи, якщо створилася виробнича ситуація, небезпечна для його життя чи здоров'я або людей, які його оточують, і навколишнього середовища.

#### 4.1.1 Правові та організаційні основи охорони праці

Основним організаційним напрямом у здійсненні управління в сфері охорони праці є усвідомлення пріоритету безпеки праці і підвищення соціальної відповідальності держави, і особистої відповідальності працівників.

Державна політика в галузі охорони праці визначається відповідно до Конституції України Верховною Радою України і спрямована на створення належних, безпечних і здорових умов праці, запобігання нещасним випадкам та професійним захворюванням. Відповідно до статті 3 Закону України «Про охорону праці» [1] (далі – Закону) законодавство про охорону праці складається з Закону, Кодексу законів про працю України [2], Закону України "Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності" [3] та прийнятих відповідно до них нормативно-правових актів, норм міжнародного договору (ратифіковані Конвенції і Рекомендації МОТ, директиви Європейської Ради).

На законодавчому рівні визначено такі пріоритетні напрямки з безпеки праці:

- Кожен працівник несе безпосередню відповідальність за порушення зазначених Законом, нормами і правилами вимог.
- Напрямки реалізації конституційного права громадян на їх життя і здоров'я в процесі трудової діяльності:
- Пріоритет життя і здоров'я працівників по відношенню до результатів виробничої діяльності підприємства.
- Повна відповідальність роботодавця за створення належних – безпечних і здорових умов праці.
- Соціальний захист працівників, повне відшкодування збитків особам, які потерпіли від нещасних випадків на виробництві та професійних захворювань.
- Комплексне розв'язання завдань охорони праці.
- Підвищення рівня промислової безпеки шляхом забезпечення суцільного технічного контролю за станом виробництв, технологій та продукції, а також сприяння підприємствам у створенні безпечних та нешкідливих умов праці.
- Соціальний захист працівників, повне відшкодування збитків особам, які потерпіли від нещасних випадків на виробництві та професійних захворювань.
- Використання економічних методів управління охороною праці, участь держави у фінансуванні заходів щодо охорони праці.
- Використання світового досвіду організації роботи щодо поліпшення умов і підвищення безпеки праці на основі міжнародної співпраці.

Користувачі персональних комп'ютерів, для яких ця робота є головною, підлягають медичним оглядам: попереднім - під час влаштування на роботу і періодичним — протягом професійної діяльності раз на два роки. Жінок з часу встановлення вагітності та в період годування дитини грудьми до роботи з ПК не допускають.

Наявні трудові відносини між працівниками і роботодавцями в Україні за темою дипломного проекту регулюються Кодексом законів про працю (КЗпП) України, відповідно до якого права працюючої людини на охорону праці охороняються всебічно та норми охорони праці неухильно інтегровані до правил внутрішнього розпорядку організації/підприємства.

#### **4.1.2 Організаційно-технічні заходи з безпеки праці**

В організації/підприємстві проводиться навчання і перевірка знань з питань охорони праці відповідно до вимог Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці України від 26.01.2005 N 15, зареєстрованого в Міністерстві юстиції України 15.02.2005 за N 231/10511 [4].

Обов'язковими вимогами враховане наступне:

- Ознайомлення з правилами безпеки праці, одержання відповідних інструктажів засвідчується у журналі інструктажів.

- Перед допуском до самостійної роботи кожен працівник має право на навчання з питань охорони праці і роботодавець зобов'язаний, і проводить таке навчання у вигляді двох інструктажів з питань охорони праці:

- 1) Вступного, який проводять працівники служби охорони праці об'єкта господарювання з усіма працівниками, яких приймають на роботу незалежно від їхньої освіти та стажу роботи за програмою, в якій подають загальні питання охорони праці із врахуванням її особливостей на об'єкті господарювання.

- 2) Первинного, який проводять керівники структурних підрозділів на місці праці з кожним працівником до початку їхньої роботи на цьому робочому місці.

Проходження працівником цих інструктажів з питань охорони праці підтверджується записами у відповідних журналах обліку інструктажів і скріплюється підписами осіб, які проводили інструктажі та осіб, які отримали інструктажі.

- 3) Повторний (не рідше одного разу в 6 місяців);

- 4) Позаплановий (при зміні правил охорони праці);

- 5) Поточний (проводять з працівниками перед виконанням робіт, на яких оформляється наряд-допуск)

- Обов'язкові організаційні заходи перед початком, під час і після завершення роботи повинні включати перевірку (візуально) наявності і справності електрообладнання та його

заземлення, а під час виконання роботи вимогу «не залишати без нагляду обладнання, яке працює». Після закінчення роботи - вимагається прибирання робочого місця, відключення всіх електроприладів від електромережі.

## 4.2 Аналіз стану умов праці

Робота над створенням системи оптимізації запитів до бази даних проходитиме в приміщенні гуртожитку СНУ ім. В. Даля (бульвар Дружби народів 41б). Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

### 4.2.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в табл. 4.1.

Таблиця 4.1 – Розміри приміщення

| Найменування          | Значення |
|-----------------------|----------|
| Довжина, м            | 6        |
| Ширина, м             | 3        |
| Висота, м             | 2,5      |
| Площа, м <sup>2</sup> | 18       |
| Об'єм, м <sup>3</sup> | 45       |

Згідно з [5] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм - не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

Для забезпечення потрібного рівного освітленості кімната має вікно та систему загального рівномірного освітлення, що встановлена на стелі. Для дотримання вимог пожежної безпеки встановлено порошковий вогнегасник та систему автоматичної пожежної сигналізації.

### 4.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за [10] і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Приміщення кабінету має об'єм 45 м<sup>3</sup>, площу - 18 м<sup>2</sup>.

Температура в приміщенні протягом року коливається у межах 18–24°C, відносна вологість - близько 50%. Система вентилявання приміщення - природна неорганізована, а опалення - централізоване.

Таблиця 4.2 - Характеристики робочого місця

| Найменування параметра                             | Фактичне значення | Нормативне значення |
|--|-------------------|---------------------|
| Висота робочої поверхні, мм                        | 750               | 680 ÷ 800           |
| Висота простору для ніг, мм                        | 730               | не менше 600        |
| Ширина простору для ніг, мм                        | 660               | не менше 500        |
| Глибина простору для ніг, мм                       | 700               | не менше 650        |
| Висота поверхні сидіння, мм                        | 470               | 400 ÷ 500           |
| Ширина сидіння, мм                                 | 400               | не менше 400        |
| Глибина сидіння, мм                                | 400               | не менше 400        |
| Висота поверхні спинки, мм                         | 600               | не менше 300        |
| Ширина опорної поверхні спинки, мм                 | 500               | не менше 380        |
| Радіус кривини спинки в горизонтальній площині, мм | 400               | 400                 |
| Відстань від очей до екрану дисплея, мм            | 800               | 700 ÷ 800           |

Розміщення вікон забезпечує природне освітлення з коефіцієнтом природного освітлення не менше 1,5%, а загальне штучне освітлення, яке здійснюється за допомогою восьми люмінесцентних ламп, забезпечує рівень освітленості не менше 200 Лк.

За ступенем пожежної безпеки приміщення належить до категорії В.

#### 4.2.3 Навантаження та напруженість процесу праці

За фізичним навантаженням робота відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни. За ступенем нервово-психічної напруги виконання роботи можна віднести до II - III ступеня і кваліфікувати як помірно напружений – напружений за умови успішного виконання поставлених завдань.

Під час виконання робіт використовують ПК та периферійні пристрої (лазерні та струменеві), що призводить до навантаження на окремі системи організму. Такі перекося у напруженні різних систем організму, що трапляються під час роботи з ПК, зокрема, значна напруженість зорового аналізатора і довготривале малорухоме положення перед екраном, не тільки не зменшують загального напруження, а навпаки, призводять до його посилення і появи стресових реакцій.

Найвні психофізіологічні небезпечні та шкідливі фактори:

а) фізичного перевантаження:

- Статичного.

- Динамічного.

б) нервово-психічного перевантаження:

- Розумового перенапруження.
- Монотонності праці.
- Перенапруження аналізаторів.
- Емоційних перевантажень.

Роботу за дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви тривалістю 15 хв через кожну годину роботи.

### **4.3 Виробнича санітарія**

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

#### **4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу**

Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання НПАОП 0.00-7.15-18 [9] «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання, зокрема персональні комп'ютери (ПК) та периферійні пристрої.

Основними робочими характеристиками персонального комп'ютера є наступні:

- Робоча напруга  $U = +220\text{В} \pm 5\%$ .
- Робочий струм  $I = 2\text{А}$ .
- Споживана потужність  $P = 350\text{Вт}$ .

Робоче місце має відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [6].

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.3).



Таблиця 4.3 – Аналіз небезпечних і шкідливих виробничих факторів

| Небезпечні і шкідливі виробничі фактори   | Джерела факторів (види робіт)   | Кількіс на оцінка | Нормативні документи |
|---|---|-------------------|----------------------|
| 1   | 2   | 3                 | 4                    |
| <b>Фізичні</b>  |   |                   |                      |
| - підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини | -//-  | 4                 | [7]                  |
| - недостатність природного світла   | порушення умов праці (вимог до приміщень)   | 2                 | [8]                  |
| - недостатнє освітлення робочої зони  | порушення гігієнічних параметрів виробничого середовища   | 3                 | [8]                  |
| <b>Психофізіологічні:</b>   |   |                   |                      |
| - нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)               | - пошук інформації для постановки теми;<br>- пошук та аналіз аналогів і літератури;<br>- пошук наявних технологій, моделювання та аналіз алгоритмів;<br>- виконання роботи за темою диплома, тестування;<br>- оформлення роботи | 4                 | [9]<br>[6]           |
| - фізичні (статичне – сидіння)  | порушення умов праці (організації місця праці- сидіння користувача, ) та організації робочого часу - безпервна робота)  | 2                 | [9]<br>[6]           |

#### 4.3.2 Пожежна безпека

Приміщення оснащено системою автоматичної пожежної сигналізації, має 1 вогнегасник ВП-5 із зарядом вогнегасної речовини 8-12 кг, відповідно до вимог чинного законодавства України. Проходи до засобів пожежогасіння вільні, не захарашуються та у разі потреби забезпечувати евакуацію всіх людей, які перебувають у приміщенні через один евакуаційний вихід з дверима на шляху евакуації, що відчиняється в напрямку виходу з будівлі від робочого місця. В приміщенні наявна затверджена «План-схема евакуації з кабінету (приміщення)».

Пожежна безпека при застосуванні ЕОМ забезпечується:

- 1) Системою запобігання пожежі.
- 2) Системою протипожежного захисту.
- 3) Організаційно-технічними заходами.

Згідно ДБН В.2.5-28:2018 [8] таке приміщення, площею 25 м<sup>2</sup>, відноситься до категорії

"В" (пожежонебезпечної) та для протипожежного захисту в ньому проектом передбачено устаткування автоматичною пожежною сигналізацією із застосуванням датчиків-сповіщувачів РІД-1 (сповіщувач димовий ізоляційний) в кількості 1 шт., і застосуванням первинних засобів пожежогасіння.

Горючими матеріалами в приміщенні, де розташовані ЕОМ, є:

- 1) Поліамід – матеріал корпусу мікросхем, горюча речовина, температура самозаймання 420° С.
- 2) Полівінілхлорид – ізоляційний матеріал, горюча речовина, температура запалювання 335° С, температура самозаймання 530° С.
- 3) Склотекстоліт ДЦ – матеріал друкарських плат, важкогорючий матеріал, показник горючості 1.7А, не схильний до температурного самозаймання.
- 4) Пластикат кабельний №.489 – матеріал ізоляції кабелів, горючий матеріал, показник горючості більше 2.1.
- 5) Деревина – будівельний і обробний матеріал, з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, температура запалювання 255° С, температура самозаймання 399° С.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол.

#### **4.4 Гігієнічні вимоги до параметрів виробничого середовища. Освітлення**

Збільшення освітленості сприяє поліпшенню працездатності навіть в тих випадках, коли процес праці практично не залежить від зорового сприйняття. При поганому освітленні людина швидко втомлюється, працює менш продуктивно, виникає потенційна небезпека помилкових дій і нещасних випадків.

Освітленість приміщення має велике значення при роботі на ПЕОМ. Вона багато в чому визначається колірною і мережевий обстановкою. Для зменшеного поглинання світла стеля і стіни вище панелей (1,5-1,7м.). Якщо вони не облицьовані звукопоглинальним матеріалом, фарбуються білою водоемульсійною фарбою (коефіцієнт відбиття повинен бути не менше 0,7). Для забарвлення стіни панелей рекомендується віддавати перевагу світлим фарбам.

Природне освітлення, коли робочі місця з ПЕОМ розташовуються в один ряд по довжині приміщення на відстані 0,8 - 1,0 м від стіни з віконними прорізами, і екрани знаходяться перпендикулярно цієї стіни. Основний потік природного світла при цій повинен бути зліва. Не

допускається спрямування основного світлового потоку природного світла праворуч, ззаду і спереду працює на ПЕОМ. Оптимальна відстань очей до екрана відео монітора повинна становити 60-70 см, допустиме не менше 50 см. Розглядати інформацію ближче 50 см не рекомендується.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає ДБН В.2.5-28:2018 [8]. Джерелом природного освітлення є сонячне світло. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє [6] і для даного приміщення в світлий час доби достатньо природного освітлення.

Розрахунок освітлення.

Для будівель виробництв світловий коефіцієнт приймається в межах 1/6 - 1/10:

$$\sqrt{a^2 + b^2} * S_b = (1/8 \div 1/10) * S_n \quad (4.1)$$

де  $S_b$  – площа віконних прорізів, м<sup>2</sup>;

$S_n$  – площа підлоги, м<sup>2</sup>.

$$S_n = a * b = 6 * 3 = 18 \text{ м}^2$$

$$S_{\text{вік}} = 1/8 * 18 = 2,25 \text{ м}^2$$

Приймаємо 2 вікна площею  $S = 1,13 \text{ м}^2$  кожне.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників  $n$  виробляється по формулі (4.2):

$$n = \frac{E * S * Z * K}{F * U * M} \quad (4.2)$$

де  $E$  – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

$S$  – освітлювана площа, м<sup>2</sup>;  $S = 18 \text{ м}^2$ ;

$Z$  – поправочний коефіцієнт світильника ( $Z = 1,15$  для ламп розжарювання та ДРЛ;  $Z = 1,1$  для люмінесцентних ламп) приймаємо рівним 1,1;

$K$  – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

$U$  – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

$M$  – число люмінесцентних ламп в світильнику – 2;

$F$  – світловий потік лампи – 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 * 18 * 1,1 * 1,5}{5400 * 0,575 * 2} \approx 1$$

Приймаємо освітлювальну установку, яка складається з одного світильника, який складається з люмінесцентної лампи загальною потужністю 160 Вт, напругою – 220 В.

#### 4.5 Вентилювання

Здійснюється провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

#### 4.6 Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Загальний опір захисного заземлення визначається за формулою:

$$R_{\text{зн}} = \frac{R_3 * R_n}{R_n * n * \eta_3 + R_3 * n_n} \quad (4.3)$$

де  $R_3$  - опір заземлення, якими когут бать труби, опори, кути і т.п., Ом;

$R_{\text{ш}}$  - опір опори, яке з'єднує заземлювачі, Ом;

$n$  - кількість заземлювачів;

$\eta_3$  - коефіцієнт екранування заземлювача; приймається в межах  $0,2 \div 0,9$ ;  $\eta_3 = 0,7$

$\eta_{\text{ш}}$  - коефіцієнт екранування сполучної стійки; приймається в межах  $0,1 \div 0,7$ ;  $\eta_{\text{ш}} = 0,5$ ;

Опір заземлення визначається за формулою:

$$R_3 = \frac{\rho}{2\pi \cdot l} \cdot \left( \ln \frac{2 \cdot l}{d} + \frac{1}{2} \ln \frac{4 \cdot t + l}{4 \cdot t - l} \right) \quad (4.4)$$

де  $\rho$  - питомий опір ґрунту, залежить від типу ґрунту, Ом·м;

для піску -  $400 \div 700$  Ом·м; приймаємо  $\rho = 400$  Ом·м;

$l$  - довжина заземлювача, м; для труб - 2-3 м;  $l = 3$  м;

$d$  - діаметр заземлювача, м; для труб - 0,03-0,05 м;  $d = 0,05$  м;

$t$  - відстань від середини забитого в ґрунт заземлювача до рівня землі, м;  $t = 2$  м.

$$R_3 = \frac{400}{2 \cdot 3,14 \cdot 3} \left( \ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \ln \frac{4 \cdot 2 + 3}{4 \cdot 2 - 3} \right) = 110, \text{ Ом}$$

Опір смуги, що з'єднує заземлювачі, визначається за формулою:

$$R_{\text{ш}} = \frac{\rho}{2\pi \cdot L} \cdot \ln \frac{2 \cdot L^2}{b \cdot t^1} \quad (4.5)$$

де  $L$  - довжина смуги, що з'єднує заземлювачі (м) і приблизно дорівнює периметру будівлі:  $P_{\text{буд.}} = 42 \cdot 2 + 38 \cdot 2 = 160$  м;  $L = 160$  м;

$b$  - ширина смуги, м;  $b = 0,03$  м;

$t_1$  - глибина заземлення від рівня землі, м;  $t_1 = 0,5$  м.

$$R_n = \frac{400}{2 \cdot 3,14 \cdot 160} \cdot \ln \frac{2 \cdot 160^2}{0,03 \cdot 0,5} = 5,99, \text{ Ом}$$

Кількість заземлювачів захисного заземлення визначається за формулою:

$$n = \frac{2 \cdot R_3}{4 \cdot \eta_3} \quad (4.6)$$

де 4 - допустимий загальний опір, Ом;

2 - коефіцієнт сезонності.

Визначаємо загальний опір захисного заземлення:

$$R_{ззп} = \frac{110 \cdot 5,99}{5,99 \cdot 79 \cdot 0,7 + 110 \cdot 0,5} = 1,7 \text{ Ом}$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова:  $R_{ззп} < 4 \text{ Ом}$ .

3) При виникненню пожеж при роботі на ПЕОМ від таких можливими джерел запалювання як:

- Іскри і дуги коротких замикань.
- Перегрів провідників, резисторів та інших радіодеталей ПЕОМ, від тривалої перевантаження та наявності перехідного опору.
- Іскри при розмиканні і розмиканні ланцюгів.
- Розряди статичної електрики.
- Необережному поводженню з вогнем, а також вибухи газо-повітряних і паро-повітряних сумішей.

#### 4.7 Екологія

Діяльність за темою магістерської роботи, а саме: оптимізація запитів до бази даних в процесі її виконання впливає на навколишнє природне середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища» [11], Законом України «Про забезпечення санітарного та епідемічного благополуччя населення» [12], Законом України «Про відходи» [13].

В процесі діяльності з виконанням дипломного проектування виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи: відпрацьовані люмінесцентні лампи - I клас небезпеки, змінні носії інформації - IV клас небезпеки, відпрацьовані вогнегасники - IV клас небезпеки, макулатура - IV клас небезпеки.

## Висновки до розділу 4

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Було встановлено, що в процесі діяльності з виконанням дипломного проектуванням виникають процеси поводження з відходами ІТ галузі, які мають вплив на екологію. Серед цих відходів слід виділити:

- Відпрацьовані люмінесцентні лампи - I клас небезпеки.
- Змінні носії інформації - IV клас небезпеки.
- Відпрацьовані вогнегасники - IV клас небезпеки.
- Макулатура - IV клас небезпеки.

Завданням даної магістерської роботи було оптимізувати запити до бази даних, і як результат було проаналізовано методи оптимізації запитів. Згідно з аналізу в подальшому розроблятиметься реальна система, яка значно полегшить процес обробки запитів до бази даних. Так як в процесі проектування використовувалося програмне забезпечення від Oracle, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для умов праці з використанням персонального комп'ютера на якому будуть аналізуватися методи для оптимізації запитів.

## Література до четвертого розділу

1. Закон України "Про охорону праці". Вводиться в дію Постановою ВР № 2695-ХІІ від 14.10.92, ВВР, 1992, № 49, ст. 669. - Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/2694-12](http://www.URL:https://zakon.rada.gov.ua/laws/show/2694-12)
2. Кодекс законів про працю України. Затверджується Законом № 322-VІІІ від 10.12.71 ВВР, 1971. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/322-08](http://www.URL:https://zakon.rada.gov.ua/laws/show/322-08)
3. Закон України "Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності". Наказ від 21 грудня 2000 року N 2180-ІІІ. Режим доступу: [www. URL: https://dnaop.com/html/2065/doc-zakon-ukrajini-pro-zagalynoobovjzskove-derzhavne-socialyne-strahuvannya-vid-neshhasnogo-vipadku-na-virobnictvi-ta-profesijnogo-z](http://www.URL:https://dnaop.com/html/2065/doc-zakon-ukrajini-pro-zagalynoobovjzskove-derzhavne-socialyne-strahuvannya-vid-neshhasnogo-vipadku-na-virobnictvi-ta-profesijnogo-z)
4. Про затвердження Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці (НПАОП 0.00-4.12-05). Наказ від 26.01.2005 №15. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/z0231-05](http://www.URL:https://zakon.rada.gov.ua/laws/show/z0231-05)
5. Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. Постанова N 42 від 01.12.99. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/va042282-99](http://www.URL:https://zakon.rada.gov.ua/rada/show/va042282-99)
6. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98. Затверджено Постановою Головного державного санітарного лікаря України 10 грудня 1998 р. N 7. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/v0007282-98](http://www.URL:https://zakon.rada.gov.ua/rada/show/v0007282-98)
7. Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом. Наказ від 1 липня 2016 року N 204. Режим доступу: [www. URL: http://epicentre.co.ua/dstu/doc28522.html](http://www.URL:http://epicentre.co.ua/dstu/doc28522.html)
8. ДБН В.2.5-28:2018 «Природне і штучне освітлення». Режим доступу: [www. URL: http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf](http://www.URL:http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf)
9. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за № 508/31960. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/z0508-18](http://www.URL:https://zakon.rada.gov.ua/laws/show/z0508-18)
10. ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою». Наказ від 15.06.2016 №158. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/v0158858-16](http://www.URL:https://zakon.rada.gov.ua/rada/show/v0158858-16)
11. Закон України «Про охорону навколишнього природного середовища» . Вводиться в дію Постановою ВР № 1268-ХІІ від 26.06.91, ВВР, 1991, № 41, ст.547. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/1264-12](http://www.URL:https://zakon.rada.gov.ua/laws/show/1264-12)
12. Закони України «Про охорону навколишнього природного середовища». Вводиться в

дію Постановою ВР № 4005-ХІІ від 24.02.94, ВВР, 1994, № 27, ст.219. Режим доступу: [www.  
URL: https://zakon.rada.gov.ua/laws/show/4004-12](http://www.url:https://zakon.rada.gov.ua/laws/show/4004-12)

13. Закон України «Про відходи». Відомості Верховної Ради України (ВВР), 1998, № 36-37, ст. 242. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/187/98-вр](http://www.url:https://zakon.rada.gov.ua/laws/show/187/98-вр)



## ВИСНОВКИ

У першому розділі були розглянуті такі програмні засоби (ПЗ), як: SQL Sentry, System Center Operation Manager, Idera DB Optimizer Product, Quest Software, Oracle Enterprise Manager. Всі ці ПЗ спрямовані на моніторинг станів бази даних. Моніторинг є корисним засобом, завдяки якому можна знайти проблеми на ранніх стадіях і швидко їх вирішити. Серед переваг використання ПЗ можна виділити такі функції:

- Масштабування для підтримки моніторингу.
- Аналіз виконання SQL запиту.
- Експертні рекомендації.
- Виявлення аномалій завдяки машинному навчанню.
- Підтримка оптимізації.

З використанням ПЗ можуть виникнути труднощі з несвоєчасним реагування на критичні стани бази даних/середовища, малою кількістю інформації для прийняття рішення, виявленні аномалій.

Несвоєчасне реагування на критичні стани бази даних/середовища зменшує продуктивність бази даних та є загроза втрати даних. В режимі реального часу немає можливості швидко впливати на ситуацію та вирішувати критичні проблеми.

Більшість ПЗ не надають повної інформації про виникнення проблем в базі даних. Звіти про проблеми не завжди можуть надати повне уявлення про проблему та її першопричини. Важливо розуміти всі причини виникнення проблеми та можливість в майбутньому її відстежити/вирішити на основі отриманої інформації.

Програмні засоби не визначають відхилення параметрів від їх стану по замовчуванням. В разі виникнення аномалій, їх не можна буде виявити та виправити.

Було прийнято рішення про створення механізму для своєчасного оповіщення та надання інформації для оптимізації запиту до бази даних. Механізм не буде мати тих недоліків, які є в розглянутих ПЗ.

В другому розділі розглянуті системні метрики, було обрано математичний метод та розраховано ймовірності оптимізації запитів.

Під час виконання запиту можуть стати непередбачувані стани системи. Використання системних ресурсів, неправильна побудова схеми бази даних, складність запиту – фактори, які впливають на зниження продуктивності. Для виявлення проблем на боці бази даних можна використовувати метрики стану. За допомогою метрик можна переконатися в тому, що система перебуває не в продуктивному стані. Тобто не використовується максимальна продуктивність системи. Велика кількість зчитувань даних впливає на простоті процесору та навпаки, під час обробки даних процесору формується черга на виконання. За допомогою журналу повільних

запитів можна виявити повільні запити та з'ясувати час виконання цих запитів, кількість перевірених рядків під час виконання запитів, загальну кількість блокувань. Щоб кожного разу не слідкувати за показниками метрик, було здійснено злиття даних метрик за допомогою підрахованих ймовірностей за допомогою ВВА (basic belief assignment). За отриманим значенням злиття було зроблено висновок, що систему можна оптимізувати методами оптимізації. Була розрахована ймовірність оптимізації кожного методу та виявлено, що метод індексування має найбільшу ймовірність для оптимізації запиту.

Якщо метрики не сигналізують про навантаження системи, а запит виконується повільно, треба шукати проблеми в самому запиті. Складну структуру запиту можна розділити на прості під запити та таким чином слідкувати за показниками системи.

Щоб не перевіряти метрики, були розраховані приблизні ймовірності для розглянутих методів оптимізації. Серед розглянутих методів оптимізації найкращі показники має індексування. Індексування прискорює час обробки запитів та використовує менше всіх системних ресурсів.

У третьому розділі описано етапи розробки методики та фіксування показників бази даних під час обробки запитів.

Для фіксування повільних запитів та сповіщення користувача про перевищення порогових значень метрик буда створена методика. Розроблена методика відображає стан системи під час обробки запиту. Вона фіксує:

- Кількість повільних запитів.
- Кількість таблиць, з яких відбувається зчитування.
- Середню кількість оброблених запитів в секунду.
- Кількість потоків, які обробляють запити.

Методика аналізує:

- Показники не оптимізованих запитів.
- Показники запитів з використанням методу індексування.
- Показники запитів з використанням секціонування.

Методика сповіщає:

- Про перевищення пікових значень метрик.
- Про час обробки запитів.

Методика допомагає фіксувати повільні записи, відстежувати стан системи та аналізувати порогові значення для обраних метрик.

Після аналізу всіх станів запитів, методика аналізує показники та надає рекомендації щодо найкращого методу оптимізації.

В четвертому розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну

праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Було встановлено, що в процесі діяльності з виконанням дипломного проектуванням виникають процеси поводження з відходами ІТ галузі, які мають вплив на екологію. Серед цих відходів слід виділити:

- Відпрацьовані люмінесцентні лампи - I клас небезпеки.
- Змінні носії інформації - IV клас небезпеки.
- Відпрацьовані вогнегасники - IV клас небезпеки.
- Макулатура - IV клас небезпеки.

Завданням даної магістерської роботи було оптимізувати запити до бази даних, і як результат було проаналізовано методи оптимізації запитів. Згідно з аналізу в подальшому розроблятиметься реальна система, яка значно полегшить процес обробки запитів до бази даних. Так як в процесі проектування використовувалося програмне забезпечення від Oracle, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для умов праці з використанням персонального комп'ютера на якому будуть аналізуватися методи для оптимізації запитів.

Додаток А  
Схема алгоритму

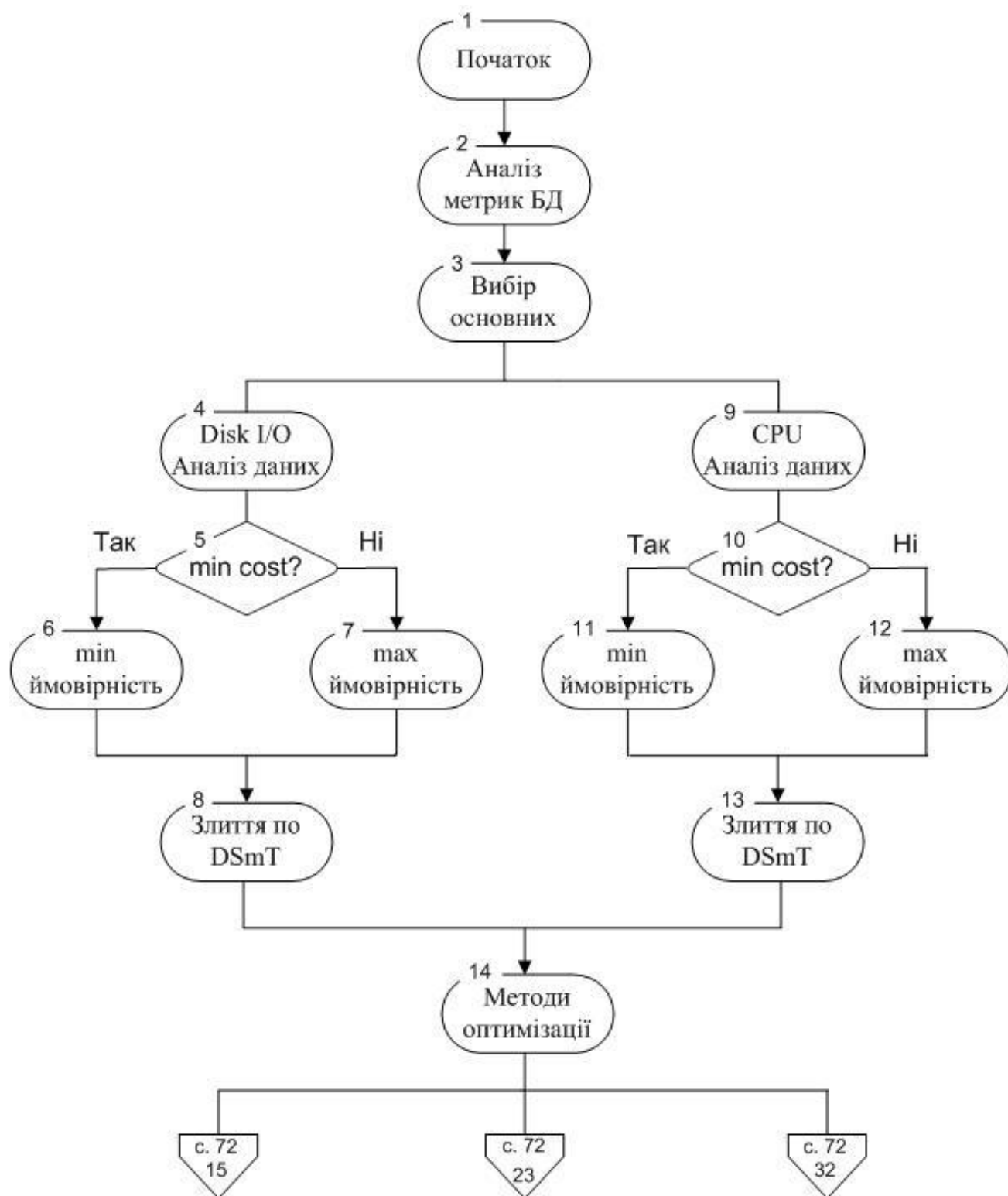


Рисунок А.1 – Схема алгоритму

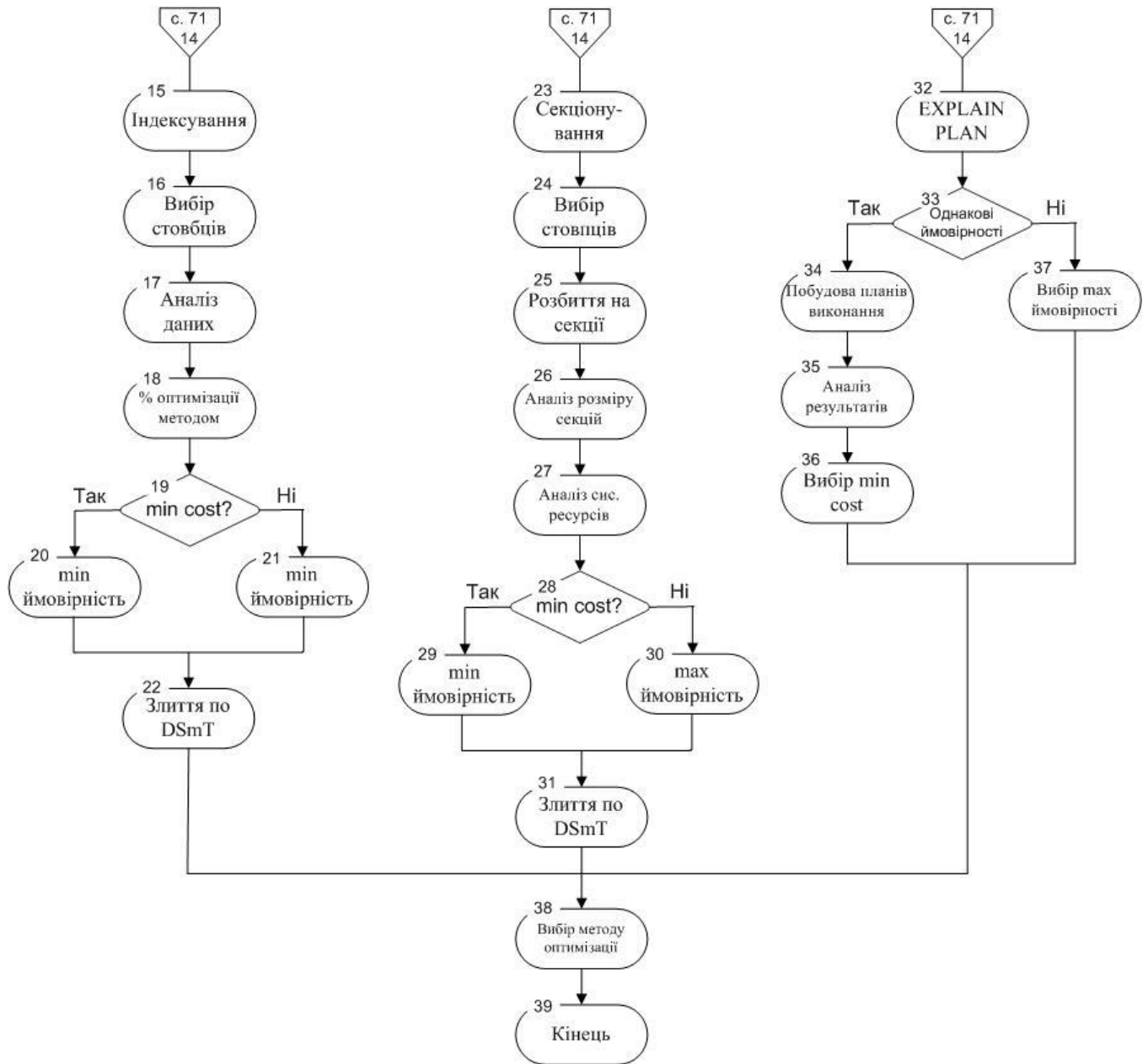


Рисунок А.2 – Продовження схеми алгоритму

## Додаток Б

### Лістинг методики

#### Формування графіків:

```

1.    include("../db.php");
2.    include("../pChart2.1.4/class/pDraw.class.php");
3.    include("../pChart2.1.4/class/pImage.class.php");
4.    include("../pChart2.1.4/class/pData.class.php");
5.    $myData = new pData();
6.    $result = mysql_query("SELECT id,value_pers FROM values_slow_index_cpu ");
7.    while (list($dt,$sum) = mysql_fetch_row($result)) {
8.        $myData->addPoints($sum,"CPU Load");
9.        $myData->addPoints($dt,"Labels");
10.   };
11.   $unique = date("Y.m.d_H.i");
12.   $gsFilename_Traffic = "cpu_".$unique.".png";
13.   $myData->setSerieDescription("Labels","Days");
14.   $myData->setAbscissa("Labels");
15.   $myData->setAxisUnit(0,"");
16.   $serieSettings = array("R"=>229,"G"=>11,"B"=>11,"Alpha"=>100);
17.   $myData->setPalette("Total",$serieSettings);
18.   $myPicture = new pImage(1250,400,$myData);
19.   $myPicture-
>setFontProperties(array("FontName"=> "../pChart2.1.4/fonts/calibri.ttf", "FontSize"=>8));
20.   $myPicture->setGraphArea(50,20,1230,380);
21.   $myPicture->drawScale();
22.   $myPicture->drawLineChart();
23.   $myPicture-
.>drawPlotChart(array("DisplayValues"=>FALSE,"PlotBorder"=>TRUE,"BorderSize"=>0,"Surroundi
ng"=>-60,"BorderAlpha"=>50));
24.   $myPicture-
>drawLegend(700,10,array("Style"=>LEGEND_NOBORDER,"Mode"=>LEGEND_HORIZONT
25.   AL));
26.   $myPicture->Render("../grapfs_cpu\\".$gsFilename_Traffic);
27.   $img = "<img src='grapfs_cpu/".$gsFilename_Traffic.'" width='100%'>";
28.   $path = 'grapfs_cpu/'.$gsFilename_Traffic;

```

29. echo \$img;

### **Визначення навантаження на центральний процесор:**

```

1. include("../db.php");
2. function get_server_load() {
3.     if (strstr(PHP_OS, 'win')) {
4.         $wmi = new COM("Winmgmts://");
5.         $server = $wmi->execquery("SELECT LoadPercentage FROM Win32_Processor");
6.         $cpu_num = 0;
7.         $load_total = 0;
8.         foreach($server as $cpu){
9.             $cpu_num++;
10.            $load_total += $cpu->loadpercentage;
11.        }
12.        return $load = round($load_total/$cpu_num);
13.    } else {
14.        $sys_load = sys_getloadavg();
15.        $load = $sys_load[0];
16.    }
17.    return $load;
18. }
19. $value_cpu = (get_server_load()/100);
20. $ins = mysql_query("INSERT INTO values_slow_index_cpu (value_pers) VALUES
('$value_cpu' ");

```

### **Визначення навантаження на зчитування/запис:**

```

1. include("../db.php");
2. $file = "F:\usr\local\mysql-5.5\data\DESKTOP-8L6BB9E-slow.log";
3. $open = file_get_contents($file);
4. //НАЙТИ ПОЗИЦИИ КОЛ-ВО СЧИТЫВАНИЙ
5. $pieces = explode(" ", $open);
6. $array_len = count($pieces);
7. $mas[]="";
8. for ($i=0;$i<$array_len;$i++) {
9.     if ($pieces[$i]>1000000) {
10.        $str = $pieces[$i];

```

```

11.  $mas[]=$str;
12.  }
13.  }
14.  //ОБРЕЗАТЬ НАЙДЕННОЕ И ОСТАВИТЬ ЛИШЬ ЦИФРОВЫЕ ЗНАЧЕНИЯ
15.  $writes[]="";
16.  foreach ($mas as $value) {
17.  $cut = substr($value, 0,-4);
18.  $writes[] = $cut;
19.  }
20.  //УДАЛЕНИЕ ОСТАТКОВ ОТ ОБРЕЗКИ
21.  unset($writes[0]);
22.  unset($writes[1]);
23.  //ПОДСЧЕТ ВЕРОЯТНОСТИ
24.  $min = min($writes);
25.  $max = max($writes);
26.  $probability_io = round(($min/$max),2);
27.  //ЗАПИСЬ В БД
28.  $ins = mysql_query("INSERT INTO values_slow_index_io (value_pers) VALUES
('$probability_io' ");

```

### **Визначення найліпшого методу оптимізації**

```

1.  include("../db.php");
2.  header("Content-Type: text/html; charset=utf-8");
3.  $NONE_CPU = mysql_query("SELECT * FROM values_slow_none_cpu");
4.  $INDEX_CPU = mysql_query("SELECT * FROM values_slow_index_cpu");
5.  $SEC_INDEX = mysql_query("SELECT * FROM values_slow_sec_cpu");
6.  if ((mysql_num_rows($NONE_CPU) == 0) OR (mysql_num_rows($INDEX_CPU) == 0) OR
(mysql_num_rows($SEC_INDEX) == 0)) {
7.  exit("Результаты пока не доступны! Завершите анализ всех состояний!");
8.  }
9.  //БЕЗ ОПТИМИЗАЦИИ
10. $non_index_cpu = mysql_query("SELECT * FROM values_slow_none_cpu WHERE
value_pers > 0.22 ");
11. $non_index_io = mysql_query("SELECT * FROM values_slow_none_io WHERE value_pers
> 0.44 ");
12. $none_cpu_count = mysql_num_rows($non_index_cpu);

```



```
13.  $none_io_count = mysql_num_rows($non_index_io);
14.  $avg_non_cpu = mysql_query("SELECT AVG(value_pers) AS AVG_LOAD FROM
values_slow_none_cpu");
15.  $avg_non_io = mysql_query("SELECT AVG(value_pers) AS AVG_LOAD FROM
values_slow_none_io");
16.  $row_avg_none_cpu = mysql_fetch_array($avg_non_cpu);
17.  $row_avg_none_io = mysql_fetch_array($avg_non_io);
18.  //ИНДЕКСИРОВАНИЕ
19.  $full_index_cpu = mysql_query("SELECT * FROM values_slow_index_cpu WHERE
value_pers > 0.22 ");
20.  $full_index_io = mysql_query("SELECT * FROM values_slow_index_io WHERE value_pers
> 0.44 ");
21.  $index_cpu_count = mysql_num_rows($full_index_cpu);
22.  $index_io_count = mysql_num_rows($full_index_io);
23.  $avg_index_cpu = mysql_query("SELECT AVG(value_pers) AS AVG_LOAD FROM
values_slow_index_cpu");
24.  $avg_index_io = mysql_query("SELECT AVG(value_pers) AS AVG_LOAD FROM
values_slow_index_io");
25.  $row_avg_index_cpu = mysql_fetch_array($avg_index_cpu);
26.  $row_avg_index_io = mysql_fetch_array($avg_index_io);
27.  //СЕКЦИОНИРОВАНИЕ
28.  $full_sec_cpu = mysql_query("SELECT * FROM values_slow_sec_cpu WHERE value_pers >
0.22 ");
29.  $full_sec_io = mysql_query("SELECT * FROM values_slow_sec_io WHERE value_pers >
0.44 ");
30.  $sec_cpu_count = mysql_num_rows($full_sec_cpu);
31.  $sec_io_count = mysql_num_rows($full_sec_io);
32.  $avg_sec_cpu = mysql_query("SELECT AVG(value_pers) AS AVG_LOAD FROM
values_slow_sec_cpu");
33.  $avg_sec_io = mysql_query("SELECT AVG(value_pers) AS AVG_LOAD FROM
values_slow_sec_io");
34.  $row_avg_sec_cpu = mysql_fetch_array($avg_sec_cpu);
35.  $row_avg_sec_io = mysql_fetch_array($avg_sec_io);
36.  //СРЕДНЕЕ ЗНАЧЕНИЕ ПРЕВЫШЕНИЯ ПОРОГА
37.  $TOTAL_AVG_NONE = ($none_cpu_count+$none_io_count)/2;
38.  $TOTAL_AVG_INDEX = ($index_cpu_count+$index_io_count)/2;
```

```

39.  $TOTAL_AVG_SEC = ($sec_cpu_count+$sec_io_count)/2;
40.  //СРЕДНЯЯ НАГРУЗКА
41.  $NZ_AVG_NONE = (round($row_avg_none_cpu['AVG_LOAD'],3) +
round($row_avg_none_io['AVG_LOAD'],3)) / 2;
42.  $NZ_AVG_INDEX = (round($row_avg_index_cpu['AVG_LOAD'],3) +
round($row_avg_index_io['AVG_LOAD'],3)) / 2;
43.  $NZ_AVG_SEC = (round($row_avg_sec_cpu['AVG_LOAD'],3) +
round($row_avg_sec_io['AVG_LOAD'],3)) / 2;
44.  //DSmT
45.  $ds_1 =
round((((($TOTAL_AVG_NONE*$NZ_AVG_NONE)+((sqrt($TOTAL_AVG_NONE)*$NZ_AVG_
NONE)/($TOTAL_AVG_NONE+$NZ_AVG_NONE))+((sqrt($NZ_AVG_NONE)*$TOTAL_AVG_
NONE)/($TOTAL_AVG_NONE+$NZ_AVG_NONE))))/100,3);
46.  $ds_2 =
round((((($TOTAL_AVG_INDEX*$NZ_AVG_INDEX)+((sqrt($TOTAL_AVG_INDEX)*$NZ_AV
G_INDEX)/($TOTAL_AVG_INDEX+$NZ_AVG_INDEX))+((sqrt($NZ_AVG_INDEX)*$TOTAL_
AVG_INDEX)/($TOTAL_AVG_INDEX+$NZ_AVG_INDEX))))/100,3);
47.  $ds_3 =
round((((($TOTAL_AVG_SEC*$NZ_AVG_SEC)+((sqrt($TOTAL_AVG_SEC)*$NZ_AVG_SEC)/
($TOTAL_AVG_SEC+$NZ_AVG_SEC))+((sqrt($NZ_AVG_SEC)*$TOTAL_AVG_SEC)/($TOTAL
_AVG_SEC+$NZ_AVG_SEC))))/100,3);
48.  $DSmT_result = min(array($ds_1,$ds_2,$ds_3));
49.  if ($DSmT_result == $ds_1) {$out_win_method = "Неоптимизированные";}
50.  if ($DSmT_result == $ds_2) {$out_win_method = "Индексируемые";}
51.  if ($DSmT_result == $ds_3) {$out_win_method = "Секционированные";}
52.  echo "
53.  <table width='100%' border='all'>
54.  <tr>
55.  <td colspan='7' align='center' style='padding:10px;font-size:16pt;'> состояния запросов</td>
56.  </tr>
57.  <tr>
58.  <td> - </td>
59.  <td colspan='2' align='center' style='padding:10px;'> Без оптимизации </td>
60.  <td colspan='2' align='center'> Индексирование </td>
61.  <td colspan='2' align='center'> Секционирование </td>
62.  </tr>

```



### Формування фонових запитів

```

1.   function run_analyze_1() {
2.     document.getElementById("st1").innerHTML='Аналіз запитів без оптимізації було
запущено! <br>';
3.     document.getElementById("st2").style.display="none";
4.     document.getElementById("st3").style.display="none";
5.     document.getElementById("ar1").innerHTML='<div class="spinner-border"
role="status"><span class="sr-only">Loading...</span></div>';
6.     metrics_none();
7.   }
8.   function run_analyze_2() {
9.     document.getElementById("st2").innerHTML='Аналіз запитів з індексуванням було
запущено! <br>';
10.    document.getElementById("st1").style.display="none";
11.    document.getElementById("st3").style.display="none";
12.    document.getElementById("ar2").innerHTML='<div class="spinner-border"
role="status"><span class="sr-only">Loading...</span></div>';
13.    metrics_index();
14.   }
15.   function run_analyze_3() {
16.     document.getElementById("st3").innerHTML='Аналіз запитів з секціонуванням було
запущено! <br>';
17.     document.getElementById("st1").style.display="none";
18.     document.getElementById("st2").style.display="none";
19.     document.getElementById("ar3").innerHTML='<div class="spinner-border"
role="status"><span class="sr-only">Loading...</span></div>';
20.     metrics_sections();
21.   }
22.   //Без оптимізації
23.   function metrics_none() {
24.     $.ajax({
25.       url: 'query_none_slow.php',
26.       success: function(data) {
27.         document.getElementById("st1").style.display="none";
28.         document.getElementById("st2").style.display="block";
29.         document.getElementById("st3").style.display="block";

```

```
30.     }
31.     });
32.     setInterval('none_metrics_values()',5000);
33.     }
34.     none_metrics_values() {
35.     $.ajax({
36.     url: 'slow/cpu_none.php',
37.     success: function(data) {
38.     document.getElementById("cpu_report").innerHTML = data;
39.     }
40.     });
41.     $.ajax({
42.     url: 'slow/io_none.php',
43.     success: function(data) {
44.     document.getElementById("io_report").innerHTML = data;
45.     }
46.     });
47.     $.ajax({
48.     url: 'write/cpu_none.php',
49.     success: function(data) {
50.     }
51.     });
52.     $.ajax({
53.     url: 'write/io_none.php',
54.     success: function(data) {
55.     }
56.     });
57.     } //setInterval('none_metrics_values()',5000);
58.     //Индексирование -> 2.Графики
59.     function metrics_index() {
60.     $.ajax({
61.     url: 'query_index_slow.php',
62.     success: function(data) {
63.     document.getElementById("st2").style.display="none";
64.     document.getElementById("st1").style.display="block";
65.     document.getElementById("st3").style.display="block";
```

```
66.     }
67.     });
68.     setInterval('metrics_index_values()',5000);
69.     }
70.     function metrics_index_values(){
71.     $.ajax({
72.     url: 'slow/cpu_index.php',
73.     success: function(data) {
74.     document.getElementById("cpu_report").innerHTML = data;
75.     }
76.     });
77.     $.ajax({
78.     url: 'slow/io_index.php',
79.     success: function(data) {
80.     document.getElementById("io_report").innerHTML = data;
81.     }
82.     });
83.     $.ajax({
84.     url: 'write/cpu_index.php',
85.     success: function(data) {
86.     }
87.     });
88.     $.ajax({
89.     url: 'write/io_index.php',
90.     success: function(data) {
91.     }
92.     });
93.     } //setInterval('metrics_index_values()',5000);
94.     //Секционирование
95.     function metrics_sections() {
96.     $.ajax({
97.     url: 'query_sec_slow.php',
98.     success: function(data) {
99.     document.getElementById("st3").style.display="none";
100.    document.getElementById("st2").style.display="block";
101.    document.getElementById("st1").style.display="block";
```

```
102. }
103. });
104. setInterval('metrics_sections_values()',5000);
105. }
106. function metrics_sections_values() {
107. $.ajax({
108. url: 'slow/cpu_sec.php',
109. success: function(data) {
110. document.getElementById("cpu_report").innerHTML = data;
111. }
112. });
113. $.ajax({
114. url: 'slow/io_sec.php',
115. success: function(data) {
116. document.getElementById("io_report").innerHTML = data;
117. }
118. });
119. $.ajax({
120. url: 'write/cpu_sec.php',
121. success: function(data) {
122. }
123. });
124. $.ajax({
125. url: 'write/io_sec.php',
126. success: function(data) {
127. }
128. });
129. } //setInterval('metrics_sections_values()',5000);
130. function delete_data_graphs() {
131. $.ajax({
132. url: 'delete_cash_data.php',
133. success: function(data) {
134. }
135. });
136. }
137. function metrics_trashhold() {
```

```
138. $.ajax({
139.   url: 'load_cpu.php',
140.   success: function(data) {
141.     document.getElementById("trashhold_cpu").innerHTML=data;
142.   }
143. });
144. $.ajax({
145.   url: 'load_io.php',
146.   success: function(data) {
147.     document.getElementById("trashhold_io").innerHTML=data;
148.   }
149. });
150. } setInterval('metrics_trashhold()',5000);
151. function show_analyze_queries() {
152.   $.ajax({
153.     url: 'show_analyze_report.php',
154.     success: function(data) {
155.       document.getElementById("place_report_methods").innerHTML=data;
156.     }
157.   });
158. }
```



## Додаток В

### Презентація

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
імені В. ДАЛЯ  
Факультет «Інформаційних технологій та електроніки»

Дипломний проект на тему:  
«Дослідження методів оптимізації запитів до бази даних»

Студент гр. КІ-18ДМ  
Бакітько Д.Е

Научний керівник:  
ст. викл. Нестеров М.В

Рисунок В.1 – Перший слайд

**Метою** дослідження є підвищення швидкості виконання запиту до бази даних з урахуванням обраного методу.

Відстеження показників системи під час обробки повільного запиту на предмет перевищення розрахованих порогових значень. Розрахунок ймовірностей навантажень на обрані метрики з використанням математичного методу для виявлення найбільш продуктивного методу оптимізації

Рисунок В.2 – Другий слайд

## Існуючі програмні засоби для моніторингу стану бази даних

1. SQL Sentry
2. System Center Operations Manager
3. Idera DB Optimizer Product
4. SolarWinds
5. Quest Software
6. Oracle Enterprise Manager

3

Рисунок В.3 – Третій слайд

| №   | Метрика  | Призначення   |
|-----|--|---|
| 1   | CPU  | Відстеження стану центрального процесору (ЦП) серверу.  |
| 2   | Storage I/O  | Відстеження операцій зчитування/запису.   |
| 3   | RAM  | Використання пам'яті для обробки запиту.  |
| 4   | Global Cache   | В випадках ідентичних запитів повертати відповідь із кешу для зберігання ресурсів та повторно не виконувати ідентичні запити. |
| 5   | Response time  | Час реакції бази даних на запити та їх обробку.   |
| 6   | Transactions per second                                    | Кількість транзакцій в секунду.   |
| 7   | Client/server bytes received by server per sec             | Кількість байтів клієнта/сервера, отриманих сервером за секунду.  |
| 8   | Deadlocks per minute                                       | Кількість блокувань (тупиків) в секунду   |
| 9   | Highest amount of permanent partition in use – MB (MEMORY) | Відсоток постійного простору розділів, який використовується.   |
| 10  | Lock Wait Duration (s)                                     | Час, в секундах, скільки агент чекав для доступу.   |
| 11  | Physical writes  | Загальна кількість блоків даних, записаних на диск.   |
| ... | ...  | ...   |

4

Рисунок В.4 – Четвертий слайд

## Обрані метрики для виявлення повільних запитів

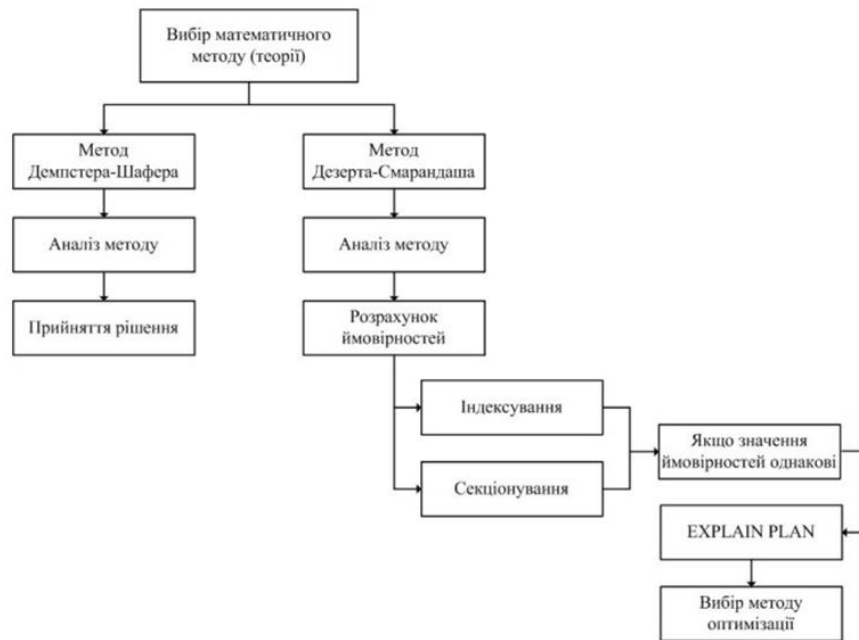
1. CPU – навантаження на центральний процесор
2. Disk I/O – навантаження на зчитування/запис
3. RAM – кількість використаної оперативної пам'яті



5

Рисунок В.5 – П'ятий слайд

## Схема вибору математичного методу та виявлення оптимального методу оптимізації



6

Рисунок В.6 – Шостий слайд

## Розрахунок ймовірностей навантаження на метрики з використанням DST

Навантаження зчитування/запису (Disk I/O) по правилу DST становить:

$$M_{DS}(X) = \frac{0.69 * 0.34}{1 - (0.69 * 0.34)} = \frac{0.2346}{0.7654} = 0.306$$

Навантаження ЦП (CPU) по правилу DST становить:

$$M_{DS}(X) = \frac{0.34 * 0.19}{1 - (0.34 * 0.19)} = \frac{0.0646}{0.9354} = 0.069$$

7

Рисунок В.7 – Сьомий слайд

## Розрахунок ймовірностей навантаження на метрики з використанням DSmT

Навантаження зчитування/запису (Disk I/O)

$$M_{PCRS}(X) = 0.69 * 0.34 + \left[ \frac{0.69^2 * 0.34}{0.69 + 0.34} + \frac{0.34^2 * 0.69}{0.34 + 0.69} \right] = 0.4676$$

Навантаження центрального процесору (CPU)

$$M_{PCRS}(X) = 0.34 * 0.19 + \left[ \frac{0.34^2 * 0.19}{0.34 + 0.19} + \frac{0.19^2 * 0.34}{0.19 + 0.34} \right] = 0.1256$$

8

Рисунок В.8 – Восьмий слайд

## Вибір математичного методу

DST представляє тільки приблизний оператор злиття, зазвичай дає невірні результати в таких ситуаціях. Таким чином, плутанина навколо обґрунтованості правила Демпстера (DST) виникає через неправильну інтерпретацію природи ситуацій, які необхідно змодельовати

Для отримання більш точних даних було прийнято рішення використовувати теорію Дезерта-Смарандаша (DSmT)

| Фокальний елемент | $m_1$ | $m_2$ | $m_{DS}$ | $m_{DSmT}$ |
|-------------------|-------|-------|----------|------------|
| A (Disk I/O)      | 0.69  | 0.19  | 0.306    | 0.4676     |
| B (CPU)           | 0.34  | 0.1   | 0.069    | 0.1256     |

9

Рисунок В.9 – Дев'ятий слайд

## Розрахунок ймовірностей індексування

Ймовірність відсутності індексування:

$$M_{PCR5}(X) = 0.2629 * 0.2745 + \left[ \frac{0.2629^2 * 0.2745}{0.2629 + 0.2745} + \frac{0.2745^2 * 0.2629}{0.2629 + 0.2745} \right] = 0.138$$

Ймовірність індексування:

$$M_{PCR5}(X) = 0.2103 * 0.2783 + \left[ \frac{0.2103^2 * 0.2783}{0.2103 + 0.2783} + \frac{0.2783^2 * 0.2103}{0.2103 + 0.2783} \right] = 0.107$$

10

Рисунок В.10 – Десятий слайд

## Порівняння теоретичних та практичних значень індексування

| Твердження                       | Практичне значення, cost | Теоретичне значення, cost |
|----------------------------------|--------------------------|---------------------------|
| Вартість запиту з індексуванням  | 146.348                  | 66                        |
| Вартість запиту без індексування | 3365.399                 | 107.477                   |

11

Рисунок В.11 – Одинадцятий слайд

## Розрахунок ймовірностей секціонування

Ймовірність секціонування

$$M_{PCR5}(X) = 0.5267 * 0.4471 + \left[ \frac{0.5267^2 * 0.4471}{0.5267 + 0.4471} + \frac{0.4471^2 * 0.5267}{0.5267 + 0.4471} \right] = 0.465$$

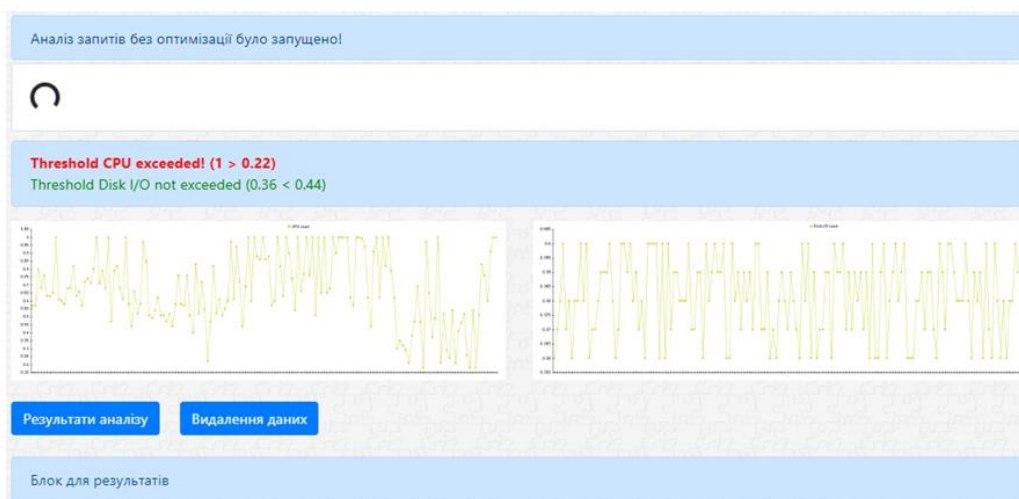
| Стан                     | m <sub>PCR5</sub> |
|--------------------------|-------------------|
| Відсутність індексування | 0.138             |
| Індексування             | 0.107             |
| Секціонування            | 0.465             |

12

Рисунок В.12 – Дванадцятий слайд

## Методика для відстежування повільних запитів

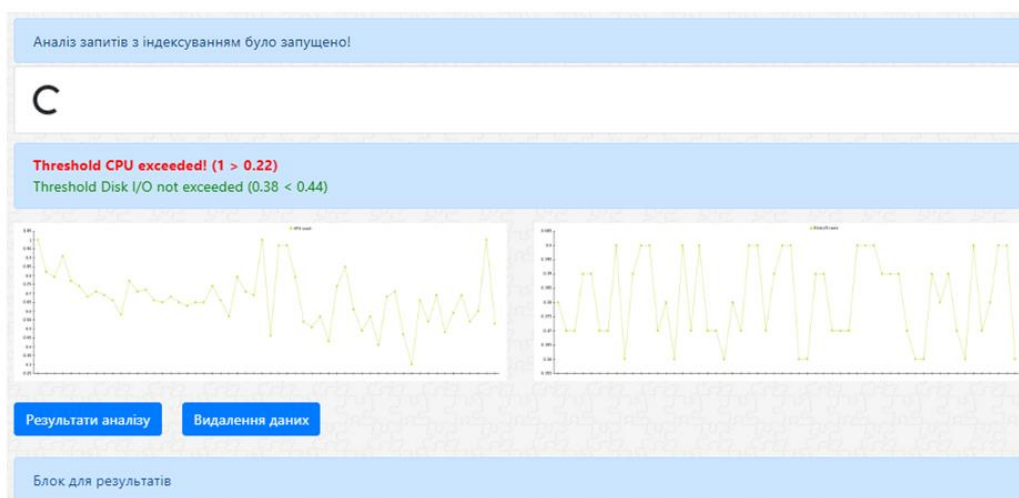
### Аналіз запитів без оптимізації



13

Рисунок В.13 – Тринадцятий слайд

### Аналіз запитів з індексуванням



14

Рисунок В.14 – Чотирнадцятий слайд

## Аналіз запитів з секціонуванням

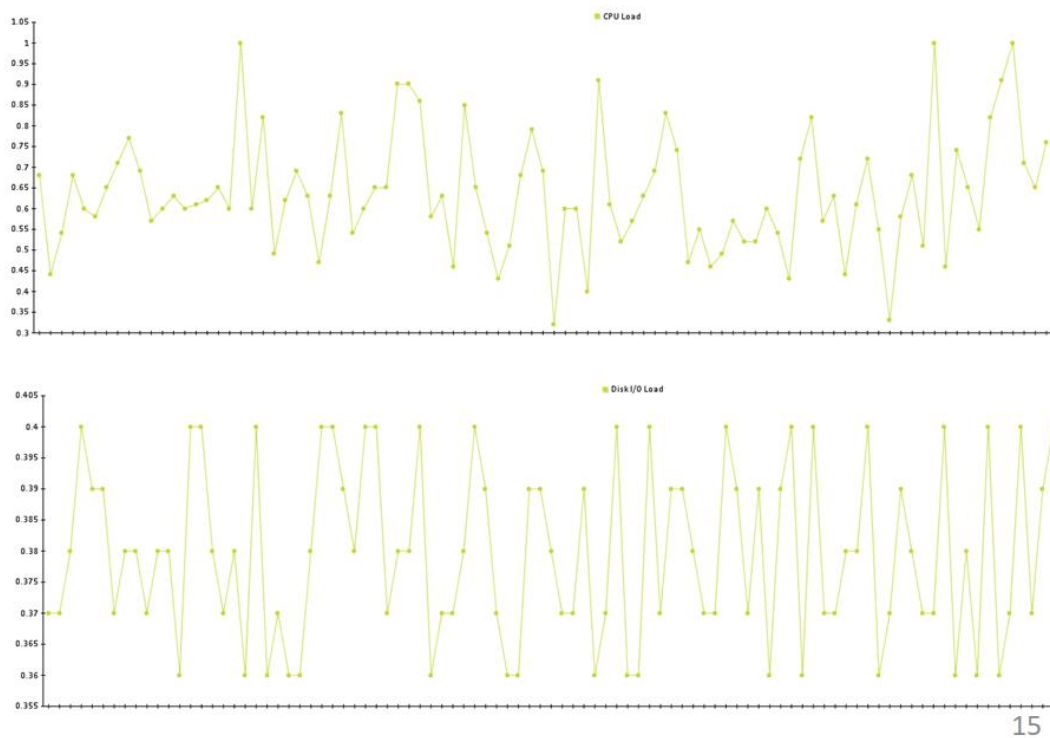


Рисунок В.15 – П'ятнадцятий слайд

## Висновки методики після аналізу методів

| Всі стани запитів              |                 |          |              |          |               |          |
|--------------------------------|-----------------|----------|--------------|----------|---------------|----------|
|                                | Без оптимізації |          | Індексування |          | Секціонування |          |
|                                | CPU             | Disk I/O | CPU          | Disk I/O | CPU           | Disk I/O |
| Перевищення порогу, кіл-ть     | 149             | 0        | 56           | 0        | 92            | 0        |
| Середнє навантаження           | 0.67            | 0.381    | 0.673        | 0.381    | 0.64          | 0.38     |
| <b>Завантаженість системи:</b> | 0.399           |          | 0.156        |          | 0.242         |          |

**Індексовані** запити найбільш ефективні. Вони мінімально перевищують порогові значення та мають менше навантаження на систему

Рисунок В.16 – Шістнадцятий слайд



## Висновки

Для аналізу стану бази під час обробки запитів обрані метрики: навантаження на зчитування/запис (Disk I/O) та навантаження на центральний процесор (CPU).

Було обрано математичний метод для розрахунку теоретичних ймовірностей навантаження на метрики. Згідно з розрахунків було встановлені порогові значення, які не мають перевищувати обрані метрики.

З використанням математичних моделей були встановлені ймовірності методів оптимізації запитів. Було встановлено, що ймовірності двох методів приблизно однакові. В такому випадку необхідно орієнтуватись на показники вартості використання системних ресурсів та використання планів виконання запитів. Згідно з плану виконання запитів було встановлено, що один з методів використовує менше системних ресурсів.

Для фіксування повільних запитів та сповіщення користувача про перевищення порогових значень метрик буда створена методика. Методика аналізує:

- Показники не оптимізованих запитів.
- Показники запитів з використанням методу індексування.
- Показники запитів з використанням секціонування.

Методика сповіщає:

- Про перевищення пікових значень метрик.
- Про час обробки запитів.

Методика допомагає фіксувати повільні записи, відстежувати стан системи та аналізувати порогові значення для обраних метрик.

Після аналізу всіх станів запитів, методика аналізує показники та надає рекомендації щодо найкращого методу оптимізації.

17

Рисунок В.17 – Сімнадцятий слайд

Дякую  
за увагу

18

Рисунок В.18 – Вісімнадцятий слайд