

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Програмної інженерії  
(повна назва)

## **АТЕСТАЦІЙНА РОБОТА** **Пояснювальна записка**

рівень вищої освіти - другий (магістерський)

Дослідження методів оптимізації баз даних програмних систем  
(тема)

Виконав: студент 2 курсу, групи ПЗСм-18-1  
Кравець М.В.  
(прізвище, ініціали)

спеціальності 121- Інженерія програмного забезпечення  
(код і повна назва спеціальності)

Освітньо-професійної програми  
(тип програми)

Програмне забезпечення систем  
(повна назва освітньої програми)

Керівник доц. Назаров О.С.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. \_\_\_\_\_

З.В.Дудар

2019 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти - другий (магістерський)

Спеціальність 121-Інженерія програмного забезпечення

(код і повна назва)

Тип програми освітньо-професійна програма

Освітня програма Програмне забезпечення систем

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

## ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Кравцю Михайлу Владиславовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів оптимізації баз даних програмних систем

затверджена наказом університету від "31" жовтня 2019 р № 1610 Ст

2. Термін подання студентом роботи до екзаменаційної комісії

17 грудня 2019 р.

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

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

## 5 Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Аналіз результатів досліджень	доц. Назаров О.С.		12.11.2019

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз стану розв'язання проблеми	25 жовтня 2019р.	
2.	Вибір методів дослідження	30 жовтня 2019р.	
3.	Опис проведених досліджень	10 листопада 2019р.	
4.	Аналіз результатів досліджень	12 листопада 2019р.	
5.	Підготовка пояснювальної записки	15 листопада 2019р.	
6.	Підготовка презентації та доповіді	1 грудня 2019р.	
7.	Попередній захист	10 грудня 2019р.	
8.	Нормоконтроль, рецензування	11 грудня 2019р.	
9.	Занесення диплома в електронний архів	13 грудня 2019р.	
10.	Допуск до захисту у зав. кафедри	17 грудня 2019р.	

Дата видачі завдання \_\_\_\_\_ 2019 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ доц. Назаров О.С.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ / ABSTRACT

Пояснювальна записка до атестаційної роботи магістра, 54 с., 3 табл., 18 рис., 13 джерел.

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

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

Метою роботи є порівняння методів оптимізації баз даних.

Методи розробки базуються на фреймворку Laravel мові програмування PHP.

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

MYSQL, PHP, PHPSTORM, ATTESTATION WORK, MATHEMATICAL MODELING, STATISTICAL INVESTIGATION, DISPERSION ANALYSIS, SYSTEM, COMPONENT.

The object of the study is the program system's database. The subject of the study is the methods of database optimization.

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

Development methods are based on Laravel framework and PHP programming language.

As a result, mathematical simulation and statistical methods software was implemented. The main factors influencing the behavior of the system were identified and investigated.

## ЗМІСТ

Вступ .....	6
1 Аналіз стану розв’язання проблеми.....	8
2 Вибір методів дослідження.....	10
2.1 Математичне моделювання.....	10
2.2 Класифікація моделей та стадії моделювання .....	11
2.3 Принципи моделювання.....	14
2.4 Дисперсійний аналіз та його різновиди .....	16
2.5 Опис методів дослідження .....	20
3 Опис проведених досліджень .....	21
3.1 Обрання предметної галузі та стеку технологій .....	21
3.2 Формулювання критеріїв дослідження .....	22
3.3 Створення дослідницької програмної моделі .....	23
3.4 Опис проведених експериментів.....	29
4 Аналіз результатів досліджень .....	32
4.1 Проведення дисперсійного аналізу .....	32
4.2 Аналіз результатів експерименту.....	35
4.3 Практичне застосування результатів дослідження .....	35
Висновки .....	38
Перелік джерел посилання .....	39
Додаток А Повна схема бази даних.....	40
Додаток Б Апробація результатів роботи.....	41
Додаток В Слайди презентації .....	43
Додаток Г Відгук та рецензії .....	52

## ВСТУП

База даних (надалі БД) – сукупність даних, що структуровані згідно з певною концепцією, яка надає інформацію про їх характеристики та взаємозв'язки між елементами. В загальному випадку база даних містить такі складові, як таблиці, збережені процедури, тригери, подання, індекси та інші об'єкти. Система керування базами даних (надалі СКБД) являє собою сукупність програмних засобів, що надають користувачеві змогу взаємодіяти з компонентами баз даних. [1]

З моменту створення перших БД, перед їх розробниками час від часу ставало питання оптимізації програмного коду як у площині ресурсних витрат, так і у питанні підвищення швидкості його роботи. По мірі зростання об'ємів даних, що потребували зберігання, вдосконалювалися і технології структурованого зберігання інформації, але ж проблема оптимізації швидкості роботи залишалася та іноді навіть ставала ще більш гостро внаслідок потреби у зберіганні колосальних об'ємів даних. До того ж, передбачалося, що системи керування базами даних будуть виконувати роль готового програмного забезпечення і заборонятимуть втручання у власний програмний код, отже вважається, що оптимізація швидкості їх роботи може виконуватись виключно шляхом використання стандартних засобів, що надає користувачеві СКБД, а також їх комбінацій. Отже, актуальність проблеми оптимізації швидкості обробки запиту на рівні СКБД не тільки не спадає, але й постійно зростає по мірі удосконалювання технологій.

Метою даної роботи у першу чергу є знаходження оптимального набору змін конфігурації та структури реляційної бази даних програмної системи певного типу для забезпечення зростання швидкості обробки запиту. Серед задач, які треба вирішити при проведенні даного дослідження найважливішими є:

– виділення факторів, що можуть впливати на швидкість обробки запиту до бази даних, а також їх трансформація у аспекти оптимізації.

- проведення дослідів відносно впливу даних факторів та їх комбінацій на загальні показники швидкості.

- узагальнення результатів експериментів за допомогою створення математичної моделі.

- формулювання рекомендацій щодо оптимізації баз даних для створення можливості їх практичного застосування, або ж використання у подальших дослідженнях.

У якості об'єкта дослідження у даній роботі представлена база даних типової структури, яка часто використовується при розробці інтернет-магазинів, отже результат проведеного дослідження може бути використаним при появі необхідності оптимізації бази даних при розробці подібних систем.

Предметом дослідження є час обробки запиту до бази даних на рівні СКБД та його оптимізація за допомогою застосування факторів, що були виділені раніше. За допомогою набору експериментів та математичного моделювання у якості методів дослідження, вдалося отримати набір експериментальних даних та узагальнити його для знаходження чіткої закономірності у залежності предмета дослідження від відповідних факторів впливу.

Результатами даного дослідження є досить чіткі числові дані, які відображають ступінь вагомості кожного з факторів впливу на кінцевий результат у вигляді часу. Ці дані можуть бути використані як у якості набору рекомендацій щодо проектування баз даних програмних систем, що подібні до розглянутої, так і у якості початкових даних для проведення нових досліджень у подібній, або ж суміжній галузі.

Результати даного дослідження опубліковані у науковому збірнику конференції MEICS-2019.

## 1 АНАЛІЗ СТАНУ РОЗВ'ЯЗАННЯ ПРОБЛЕМИ

По мірі вдосконалення технологій та зростання потреб у обробці та зберіганні даних, все більш гостро стає питання швидкості роботи подібних систем. За останні 10 років об'єми промислових дискових накопичувачів збільшилися у декілька разів та продовжують постійно зростати через потребу у зберіганні великих об'ємів даних. Але через багатократне збільшення об'ємів даних, що зберігаються, постійно зростає потреба і у вдосконаленні систем обробки.

З точки зору користувача, система повинна мати однакову поведінку в незалежності від об'єму даних, що вона обробляє. Найчастіше під системами для обробки та зберігання даних мають на увазі системи, що взаємодіють насамперед із набором структурованої інформації, яка найчастіше зберігається у різноманітних базах даних. На даний момент найрозповсюдженішим їх типом є реляційні бази даних, які мають табличну структуру зберігання даних. Кожна таблиця відповідно пов'язана з певною сутністю у концепції конкретної бази даних. Цей тип займає приблизно 80% світового ринку, але у той же час використання реляційного типу для вирішення задач зберігання великих об'ємів даних стає проблематичним. Основною перевагою реляційних баз даних є структурованість даних, що зберігаються в них, а також особливості архітектури, що направлені на підтримку цілісності даних. Дана особливість досягається завдяки зберіганню зв'язків між елементами даних, але зберігання та перевірка цих зв'язків теж потребує певних витрат, що ускладнює роботу при значних об'ємах даних та слабій структурованості.

Дослідження систем, що містять великі об'єми даних (понад 500 млн записів), що були представлені Oracle, показують, що для баз даних MySQL, виконання команди "INSERT" для запису понад 350 тисяч рядків займає приблизно 6 хвилин та 21 секунду, а виконання запиту на зчитування за допомогою команди "SELECT" для баз, що містять понад 2 млн записів може займати до 4 хвилин 10 секунд [6]. Дані заміри свідчать про те, що

використання баз даних на основі SQL, які містять великі об'єми даних, ускладнюється і такі бази даних можуть потребувати оптимізації за показником швидкості обробки запиту.

Чітких універсальних рекомендацій щодо прискорення роботи СКБД не існує через унікальність кожної бази даних та контексту її використання, але ж з поправкою на область застосування бази даних, думки спеціалістів щодо методів, які слід застосовувати, збігаються.

До цього списку можна віднести:

- додання індексів до деяких стовпчиків таблиць;
- додавання дублюючих стовпчиків;
- горизонтальне та вертикальне партиціонування таблиць;
- денормалізація бази даних.

Як не дивно, на даний момент досить складно знайти результати досліджень, що свідчили би про ефективність того чи іншого з перерахованих методів оптимізації, що стосувалися б до універсальної ситуації. Підсумовуючи результати досліджень, що є у відкритому доступі, можна зробити висновок, що універсальних рекомендацій з розбором ефективності кожного з методів не існує зовсім. Це може бути пов'язаним з тим, що ефективність одних і тих самих методів оптимізації може кардинально відрізнятись при різних контекстах використання, які залежать від багатьох факторів. У тому числі структури бази даних, розміру таблиць, типів даних окремих стовпчиків, кількості рядків у окремих таблицях тощо. Але ж паралельно з цим не втрачає актуальності й потреба у більш детальних дослідженнях методів оптимізації баз даних стосовно більш конкретних випадків їх використання. Саме таке дослідження може мати найбільшу цінність через можливість його застосування на практиці при проектуванні баз даних типових програмних систем. Тому в даному дослідженні було вирішено зосередитися на конкретному контексті використання бази даних.

## 2 ВИБІР МЕТОДІВ ДОСЛІДЖЕННЯ

### 2.1 Математичне моделювання

Математичне моделювання є методом розв'язування задачі синтезу, або аналізу певної складної системи, який ґрунтується на використанні математичної моделі системи для її дослідження. Також термін "математичне моделювання" може трактуватися як процес створення та у подальшому дослідження математичної моделі системи для відтворення та вивчення її певних властивостей, а також перенесення результатів дослідження на початкову систему за допомогою використання математичних методів. У науковій літературі цей термін досить часто використовується у більш вузькому значенні. У такому випадку під математичним моделюванням мають на увазі лише метод дослідження моделі, або її побудови.

Модель являє собою штучно створений об'єкт, що є аналогічним до об'єкта або процесу, який повинен бути дослідженим, та відтворює структуру, властивості досліджуваного об'єкта у необхідному обсязі для отримання можливості його вивчення у ситуації, коли вивчення об'єкту безпосередньо є неможливим, або ж є пов'язаним з труднощами концептуального або фінансового характеру. Створення моделі полегшує процес дослідження об'єкту. Об'єкт або процес, відносно якого створюється модель, називається прототипом.

Найчастіше розрізняють три типи моделей:

- істотно-математичні;
- логіко-математичні;
- фізичні.

Істотно-математичні моделі можуть мати фізичну природу, що може відрізнитися від прототипу, але математичний опис для моделі і прототипу є однаковим.

Логіко-математичні моделі створюються з символів. Моделі цього типу є абстрактним та будуються виключно для обчислень. Між істотно-

математичними та логіко-математичними типами моделей відсутні чіткі границі.

Фізичні моделі мають подібну до досліджуваного об'єкта природу. Така модель може відрізнятися від прототипу лише розмірами, матеріалом, або ж іншими фізичними характеристиками, що у певному контексті не будуть впливати на результати дослідження.

Чітких стандартів щодо вибору й побудови моделей не існує, але ж існує ряд умов, які варто виконувати у процесі моделювання:

- модель повинна втілювати істотні сторони прототипу;
- модель повинна бути адекватною, тобто описувати саме необхідні властивості прототипу з точністю, що необхідна для дослідження;
- при побудові повинні бути враховані особливості прототипу такі, як дискретність, динамічність, лінійність та детермінованість

## 2.2 Класифікація моделей та стадії моделювання

За основними характеристиками розрізняють наступні типи моделей:

- безперервні та дискретні;
- статичні та динамічні;
- детерміновані та імовірнісні;
- лінійні та нелінійні.

Модель можна вважати безперервною, коли тимчасові зміни, що відбуваються у ній, постійно змінюються. У разі, коли зміни можуть приймати дискретні значення – модель слід віднести до дискретного типу.

Якщо у концепції моделі ураховується час, модель можна вважати динамічною. В іншому випадку вона буде статичною

Якщо у моделі наявні показники, що є випадковими величинами або ж їх характеристиками, таку модель слід віднести до імовірнісного типу. У детермінованій моделі відсутні будь-які імовірнісні елементи.

Модель можна вважати лінійною, якщо реакція моделі на зміну зовнішніх умов є пропорційною до величини цих змін, у іншому випадку, модель буде вважатися нелінійною.

Моделювання являє собою певний процес, що складається з наступних етапів:

- побудова абстрактної моделі на основі аналізу наявної інформації й вихідної гіпотези;
- побудова моделі в конкретному (числовому) виді після проведення спостережень і експериментів.

Побудова абстрактної моделі включає:

- розбивку комплексного завдання (проблеми) на більше прості;
- пошук аналогій;
- введення символів;
- запис цільової функції й обмежень.

Комплексне завдання (проблему) доцільно розбивати на більше прості, тобто спочатку будувати окремі локальні моделі, а потім поєднувати їх у загальну — глобальну модель. Пошук аналога, тобто відомість математичного опису досліджуваного предмета, явища, процесу до відомого типу моделей, є одним з істотних моментів моделювання. Якщо математичні опис зводиться до певного типу, то це в значній мірі спрощує процес розробки моделі і її наступний аналіз. Треба однак побоюватися шаблона при пошуку аналогії. Пошук аналога вимагає відповідного вміння й навичок і є одним з неформалізованих елементів дослідження. Введення символічного запису — один з елементів моделювання. Математична модель являє собою формалізований запис, тобто запис у вигляді символів. Абстрактне моделювання закінчується звичайно записом у математичному виді цільової функції й обмежень, тобто цільова функція й обмеження одержують вираження через обрані змінні.

Побудова конкретної моделі включає:

- визначення залежностей між вхідними й вихідними параметрами об'єкта, явища, процесу;

- аналіз впливу окремих параметрів і виразно основних змінних;

- знаходження обмежень у конкретному виді;

- удосконалення й аналіз розробленої моделі.

Успіх моделювання і його цінність із погляду практичних додатків у значніше мері визначається встановленням чітких залежностей між вхідними й вихідними параметрами, що відображають основні закономірності досліджуваного предмета, явища, процесу. Якщо знайдені залежності не відображають реальних взаємозв'язків, то всі подальші результати, отримані на основі моделі; позбавлені реального змісту й будуть помилковими. Математичне моделювання здобуває особливу цінність, коли виникає необхідне навчити особливо складні процеси.[2]

В загальному випадку в процесі підготовки та проведення математичного моделювання виділяють такі етапи:

- абстрагування: зосередження на властивостях, які є загальними для багатьох об'єктів та явищ матеріального світу та нехтування відмінностями існуючими між ними;

- формулювання: вибір деякої множини засобів (символів, графічних образів тощо) для зображення абстрактних понять;

- маніпуляція: правила перетворення символічного представлення як засіб передбачення результату аналогічних маніпуляцій в реальному світі;

- аналіз: реальна інтерпретація отриманих математичних результатів з метою їх імплементації;

- аксіоматизація: строге формулювання тих властивостей, які були виведені з реального світу і які є загальними при маніпуляціях як в матеріальному світі, так і над абстрактними символами, що представляють реальний світ.

Базуючись на системному підході, може бути запропонована деяка послідовність розробки моделей, коли виділяють дві основні стадії

проектування: макропроектування та мікропроектування. На стадії макропроектування на основі даних про реальну систему  $S$  та зовнішнє середовище  $E$ , будується модель зовнішнього середовища, виявляються ресурси та обмеження для будівництва моделі системи, а також вибирається модель системи та критерії, які дозволять оцінити адекватність моделі  $M$  реальної системи  $S$ .

Побудувавши модель системи та модель зовнішнього середовища на основі критерію ефективності функціонування системи у процесі моделювання обирають оптимальну стратегію управління, що дозволяє реалізувати можливості моделі з відтворення окремих сторін функціонування реальної системи  $S$ .

### 2.3 Принципи моделювання

Модель – це штучно створений зразок у вигляді схеми, фізичних конструкцій, знакових форм або формул, який, будучи подібний до досліджуваного об'єкту (або явища), відображує і відтворює в простішому й огрубленому вигляді структуру, властивості, взаємозв'язки та стосунки між елементами цього об'єкту [3].

Принципи моделювання :

– принцип інформаційної достатності. За повної відсутності інформації про об'єкт побудувати модель неможливо. За наявності повної інформації моделювання позбавлене сенсу. Існує рівень інформаційної достатності, за досягнення якого може бути побудова на модель системи.

– принцип здійсненності. Створювана модель повинна забезпечувати досягнення поставленої мети дослідження за скінчений час.

– принцип множинності моделей. Будь-яка конкретна модель відображає лише деякі сторони реальної системи. Для повного дослідження необхідно побудувати низку моделей досліджуваного процесу, причому кожна подальша модель повинна уточнювати попередню.

– принцип системності. Досліджувана система представляється у вигляді сукупності підсистем, які взаємодіють одна з одною і моделюються стандартними математичними методами. Водночас властивості системи не є сумою властивостей її елементів.

– принцип параметризації. Деякі підсистеми модельованої системи можуть бути охарактеризовані єдиним параметром: вектором, матрицею, графіком, формулою.

Для найбільш повного виконання цієї задачі до математичної моделі висуваються такі вимоги.

Адекватність – закономірне відображення моделлю системи відповідно до цілі аналізованої задачі. Проте це не жорстка відповідність, а тільки до необхідного рівня (за числом станів, кількісною мірою тощо).

Реалізованість – відсутність протиріч і підпорядкування математичної основи моделі всім звичайним законам математичної логіки і фізичної здійснимості.

Інформативність – можливість одержання необхідних відомостей стосовно характеристик системи, які цікавлять суб'єкта.

Чутливість – можливість спостереження змін характеристик стану при зміні основних параметрів системи і зовнішніх факторів.

Еволюційність – можливість розвитку моделі і внесення в неї необхідних змін і коригувань у процесі роботи зі зміною умов задачі в міру одержання результатів.

Інваріантність у часі – можливість проведення досліджень у необхідному масштабі часу.

Універсальність – надання можливості дослідження найбільше широкого кола задач без істотного ускладнення моделі і можливості сполучення її з моделями різноманітних рівнів ієрархій.

Точність – міра якості відтворення кінцевого результату при впливі на модель і систему досліджуваного процесу. Часто наближена відповідь, що

утворюється швидше, може виявитися більш ефективною, чим більш точна відповідь, на одержання якої йде більше часу.

Економічність – забезпечення припустимого рівня вартості моделювання.

Ергономічність – простота і зручність урахування ергономічних чинників функціонування системи і проведення в процесі моделювання діалогу "дослідник - модель".

Декомпозиційність – можливість використання моделей окремих підсистем для дослідження в якості самостійних систем.

#### 2.4 Дисперсійний аналіз та його різновиди

Дисперсійний аналіз є статистичним методом, призначеним для оцінки впливу різноманітних чинників на результат експерименту, а також для виконання подальшого планування аналогічного експерименту. Цей метод дозволяє порівнювати кілька вибірок за певними ознаками, що можуть мати різноманітні значення. Загальноприйнята скорочена назва дисперсійного аналізу ANOVA (від англ. ANalysis Of VAriance).

Суть дисперсійного аналізу полягає у дослідженні значимості відмінності між середніми. Для вибірки об'єму  $n$  вибіркова дисперсія обчислюється як сума квадратів відхилень від вибіркового середнього, поділена на  $n-1$  (об'єм вибірки мінус одиниця). Таким чином, при фіксованому об'ємі вибірки  $n$  дисперсією є функція суми квадратів відхилень, що позначається для стислості,  $SS$  (від англійської - Sum of Squares – Сума Квадратів). Після обчислення дисперсії, формулюється нульова гіпотеза відносно факторів, до яких саме і відносяться вибірки. Нульова гіпотеза може бути підтвердженою, або спростованою після обчислення критерію Фішера та  $P$ -значень.

Гіпотези у дисперсійному аналізі найчастіше формулюються наступним чином:

– нульова гіпотеза: «Середні величини результативної ознаки у всіх умовах дії фактора (або градаціях фактора) однакові».

– альтернативна гіпотеза: «Середні величини результативної ознаки в різних умовах дії фактора різні».

Дисперсійний аналіз переважно застосовується для дослідження впливу декількох якісних факторів на одну значиму кількісну змінну. У його основі лежить припущення, що одні змінні можуть розглядатися у якості причин у той же час як інші – у якості наслідків. Незалежні змінні іноді називають регулюючими факторами тому, що протягом дослідження дослідник має змогу варіювати їх значення та завдяки цьому оцінювати їх вплив на кінцевий результат. У даному дослідженні у якості незалежних змінних можуть бути використані безпосередньо методи оптимізації, у той же час як час обробки запиту може виступати у якості кількісної змінної, на яку і впливають дані фактори.

Узагальнено задача дисперсійного аналізу полягає в тому, щоби за допомогою загальної варіативності виділити три окремі варіативності:

– варіативність, що зумовлена дією кожної з досліджуваних незалежних змінних.

– варіативність, що зумовлена взаємодією досліджуваних незалежних змінних.

– випадкову варіативність, що зумовлена всіма неврахованими обставинами.

Варіативність, зумовлена дією змінних, що досліджуються та їх взаємодією співвідноситься з випадковою варіативністю. Показником цього співвідношення є критерій Фішера, який найчастіше позначають як  $F$ .

$F_{\text{емп}A}$  = варіативність, що зумовлена дією змінної  $A$  / Випадкова варіативність

$F_{\text{емп}B}$  = варіативність, що зумовлена дією змінної  $B$  / Випадкова варіативність

$F_{\text{емп}}AB$  = варіативність, що зумовлена взаємодією А і Б / Випадкова варіативність

У формулу розрахунку критерію F входять оцінки дисперсій, а отже, цей метод відноситься до розряду параметричних. Чим більшою мірою варіативність ознаки обумовлена досліджуваними змінними або їх взаємодією, тим вище емпіричні значення критерію F.

На відміну від кореляційного аналізу, у дисперсійному аналізі дослідник виходить з припущення, що одні змінні виступають як впливають (іменовані факторами або незалежними змінними), а інші (результативні ознаки або залежні змінні) – схильні до впливу цих факторів. Хоча дане припущення і лежить в основі математичних процедур розрахунку, воно, проте, вимагає обережності міркувань відносно джерела та об'єкту впливу.

Дисперсійний аналіз схематично можна поділити на кілька категорій. Цей поділ здійснюється, залежно від того, скільки, по-перше, факторів бере участь в розгляді, по-друге, – скільки змінних схильні до дії факторів, і, по-третє, – по тому, як співвідносяться між собою вибірки значень. При наявності одного фактора, вплив якого досліджується, дисперсійний аналіз іменується однофакторний, і розпадається на два різновиди:

– аналіз незв'язаних (тобто різних) вибірок. Наприклад, одна група учасників експерименту вирішує завдання в умовах тиші, друга – в шумній кімнаті. (У цьому випадку, нульова гіпотеза звучала б так: «середній час вирішення завдань заданого типу буде однаковим в тиші і в шумному приміщенні», тобто не залежить від фактору шуму.)

– аналіз пов'язаних вибірок. Тобто: двох вимірів, проведених на одній і тій же групі учасників у різних умовах. Той самий приклад: в перший раз завдання вирішувалася в тиші, другий – подібна задача – в умовах шумових перешкод. (На практиці до подібних дослідів слід підходити з обережністю, оскільки можна не урахувати дію іншого фактору «навчаємість», впливу якого дослідник ризикує приписати зміни умов, а саме, шуму.)

Метод однофакторного дисперсійного аналізу застосовується в тих випадках, коли досліджуються зміни результативної ознаки (залежної змінної) під впливом зміни умов або градацій будь-якого фактора.

Впливу кожної з градацій фактора схильні різні вибірки.

Повинно бути не менше трьох градацій фактора і не менше двох спостережень в кожній градації.

Розрахунки починаються з розстановки всіх даних по стовпцях, що належать до кожного з факторів відповідно.

Наступним дією буде знаходження сум значень по стовпцях (тобто градаціях) і зведення їх в квадрат.

Фактично метод полягає в зіставленні кожної з отриманих та зведених в квадрат сум з сумою квадратів всіх значень, отриманих в усьому експерименті.

У разі, дослідження одночасного впливу двох або більшої кількості факторів, маємо справу з багатфакторним дисперсійним аналізом, який також можна поділити за типом вибірки. Якщо ж одразу декілька змінних можуть мати схильність до впливу на них декількох факторів – мова йде про багатовимірний аналіз[5].

## 2.5 Опис методів дослідження

Як було зазначено у попередньому розділі, ефективність методів оптимізації баз даних може ранжуватися у залежності від певних факторів, які у першу чергу відносяться до характеристик конкретної програмної системи. При використанні моделі імовірнісного типу, спектр впливу дослідження можна розширити, знехтувавши детермінованістю моделі. Це дасть змогу збільшити умовний показник універсальності дослідження, тим самим зробивши його більш корисним для практичного застосування.

Коли результат дослідження безпосередньо стосується часу, стає питання вибору методів дослідження, узгодженість яких з моделлю не викликала би сумнівів. Як було зазначено вище, результати дослідження

обраного явища можуть бути абсолютно непередбачуваними при зміні контексту його проведення, тому у даному випадку слід застосувати експериментальне дослідження з елементами статистичного спостереження.

Статистичне спостереження є попереднім етапом статистичного дослідження, що передбачає планомірний збір даних за певним алгоритмом для подальшого їх використання у статистичному дослідженні.

Але одного статистичного спостереження може бути замало для отримання результату дослідження, що може бути застосованим у більш широкому контексті, аніж окремий набір випадків. Оскільки дане дослідження стосується часу обробки запиту до бази даних засобами СКБД, необхідне також обчислення впливу факторів оптимізації на кінцевий результат в залежності від їх комбінації, а також встановлення ступеню їх впливу та ймовірності відтворення аналогічних результатів при застосуванні на практиці. Одним з методів, що може бути ефективно застосованим для досягнення зазначеного результату може бути дисперсійний аналіз.

### 3 ОПИС ПРОВЕДЕНИХ ДОСЛІДЖЕНЬ

#### 3.1 Обрання предметної галузі та стеку технологій

Як було наведено вище, за допомогою моделювання можна відтворити поведінку програмної системи, або її частини без використання самої системи безпосередньо. Згідно мети даної роботи, потрібно дослідити методи оптимізації бази даних програмної системи.

Згідно статистики, що надає цілісну картину відносно використання різноманітних видів баз даних, найбільш популярними на даний момент залишаються реляційні бази даних. Реляційні бази даних зберігають дані у вигляді таблиць, що можуть бути пов'язані між собою за допомогою ключів, що застосовуються безпосередньо до стовпчиків таблиці. Для прискорення здійснення вибірок, користувач реляційної бази даних як правило має змогу додавати до таблиць індекси, що мають структуру хеш-мапи, за допомогою якої значно прискорюється пошук необхідного запису по таблиці. Практично всі найпопулярніші реляційні СКБД(зокрема Oracle, MySQL, PostgreSQL) мають подібні одна до одної структури, що дає змогу виділити загальні аспекти, що можуть потребувати оптимізації та повинні бути включені до моделі.

"Оптимізація бази даних" є досить абстрактним поняттям, тому для проведення дослідження цього аспекту, потрібно з'ясувати, базу даних якого вигляду слід використовувати для дослідження. На даний момент в Україні постійно зростає ринок онлайн-торгівлі, з'являється все більше й більше інтернет-магазинів, які в свою чергу, по мірі збільшення аудиторії, потребують оптимізації для прискорення взаємодії з користувачем, а отже й збільшення конкурентоспроможності магазину. Як правило, найбільш навантаженою частиною інтернет-магазину є його каталог з набором фільтрів. Кожен товар містить у собі набір характеристик, за якими клієнт магазину має змогу фільтрувати каталог для знаходження необхідного йому товару. Отже, у якості моделі для дослідження візьмемо типову базу даних інтернет-магазину. У даному випадку дослідженню будуть підлягати таблиці,

що містять у собі інформацію про товари, типи фільтрів, та безпосередньо фільтраційні дані.

Оскільки переважна більшість інтернет-магазинів в Україні та світі створена за допомогою мови програмування PHP та різноманітних систем управління контентом, які також базуються на мові програмування PHP, у якості СКБД будемо використовувати найрозповсюдженішу у PHP-проектах MySQL.

### 3.2 Формулювання критеріїв дослідження

Для визначення факторів впливу на швидкість роботи бази даних слід з'ясувати, якого саме типу запити найчастіше використовуються відносно наведених таблиць. Як було зазначено вище, оптимізації підлягає частина системи, що відповідає за показ даних кінцевому користувачеві, а отже, будуть використовуватись переважно запити типу SELECT. Одним з найрозповсюдженіших шляхів прискорення роботи SELECT-запитів є створення індексів на полях, за якими відбувається пошук рядків у таблиці, або ж поєднання таблиць. У той же час, за відсутності індексу на полі, за яким відбувається пошук, або ж у разі поєднання таблиць у запиті (JOIN), швидкість обробки запиту буде значно знижено. Отже, позначимо наявність індексів одним із факторів впливу на швидкість обробки запиту СКБД.

Одним з найважливіших аспектів створення баз даних є їх нормалізація. Нормалізація являє собою процес перетворення структури бази даних згідно правил, які містяться у так званих нормальних формах. Основною метою нормалізації є зменшення надмірності даних зі збереженням їх цілісності. Результатом нормалізації як правило є база даних зі зміненою структурою таблиць та реорганізованою структурою зберігання даних. На перший погляд, оптимальність прямо пропорційна ступеню нормалізації, але ж у певних ситуаціях надлишкова нормалізація може значно зменшити швидкість обробки запиту всередині СКБД. Це відбувається переважно у SELECT-запитах, які використовують оператор

JOIN для приєднання таблиць. При використанні оператора JOIN або ж вкладених запитів) при зв'язуванні таблиць з відношенням "один до багатьох", або "багато до одного" відбувається кількісне помноження рядків таблиць, що призводить до сповільнення обробки запиту СКБД. У зв'язку з цим, у деяких випадках, більшу оптимальність забезпечує неповна нормалізація (використання другої нормальної форми) бази даних або її окремої частини, тому нормалізація здійснює досить значний вплив на швидкість обробки запитів до бази даних. Отже, будемо використовувати ступінь нормалізації бази даних у якості другого фактору впливу на швидкість обробки запиту на стороні СКБД.

Для спрощення поставленої задачі будемо вважати, що ці аспекти є атомарними, а також що зв'язок між ними знаходиться у межах статистичної похибки.

### 3.3 Створення дослідницької програмної моделі

Як було зазначено вище, у якості стеку технологій для дослідження було вирішено використати стандартний PHP-стек, що найчастіше використовується для створення різноманітних веб-проектів, зокрема інтернет-магазинів, інформаційних ресурсів, систем обробки та агрегації даних тощо. У даному випадку, було використано популярний MVC PHP-фреймворк Laravel. У якості СКБД був обраний MySQL.

При створенні більшості програмних продуктів незалежно від їх функціональних характеристик має місце послідовність розробки проекту. Як правило, вона складається з декількох етапів, починаючи від формулювання технічного завдання та закінчуючи тестуванням кінцевого продукту. У даному випадку можна явно виділити 4 етапи створення проекту:

- формулювання технічного завдання до проекту.
- проектування архітектури серверної частини проекту та схеми бази даних.

- реалізація технічного завдання з використанням обраного раніше стеку технологій.
- тестування коректності роботи реалізованого функціоналу.

Оскільки основною метою даної роботи є дослідження та порівняння методів оптимізації бази даних програмної системи – основним об'єктом дослідження слід вважати безпосередньо базу даних з певним змістом, який повинен відповідати реальним умовам використання. Отже, у даному випадку було вирішено реалізувати базу даних програмної системи, серверний код, що повинен її обслуговувати та веб-інтерфейс, за допомогою якого можна дослідити фактори впливу на швидкість обробки запиту на стороні СКБД шляхом внесення змін до конфігурації експериментів, бази даних та її вмісту. Решта функціоналу інтернет-магазину не може впливати на хід та результати дослідження, тому не потребує повної реалізації.

Через наявність у стеку технологій PHP-фреймворка Laravel, серверна архітектура проекту базується на архітектурному паттерні MVC, який передбачає розділення проекту на три архітектурних рівня: модель – частина, яка відповідає безпосередньо за бізнес-логіку проекту та шар доступу до даних, представлення – частина, яка відповідає за відображення інтерфейсу системи та його взаємодію з користувачем, контролер – прошарок, який виконує роль керівника та використовує два інших рівня для формування поведінки системи. Деякі розробники можуть помилково тлумачити значення рівнів даного паттерну, вважаючи, що кожному контроллеру мають відповідати лише одне відображення та одна модель.

Досить часто таке помилкове розуміння може призводити до незворотних наслідків для архітектури проекту, тому слід чітко розуміти, що розділення відбувається як правило не на рівні одного класу, або набору класів, а найчастіше на рівні просторів імен, або ж інших структурних одиниць проекту. Схематичне зображення даного архітектурного паттерну зображено на рисунку 3.1

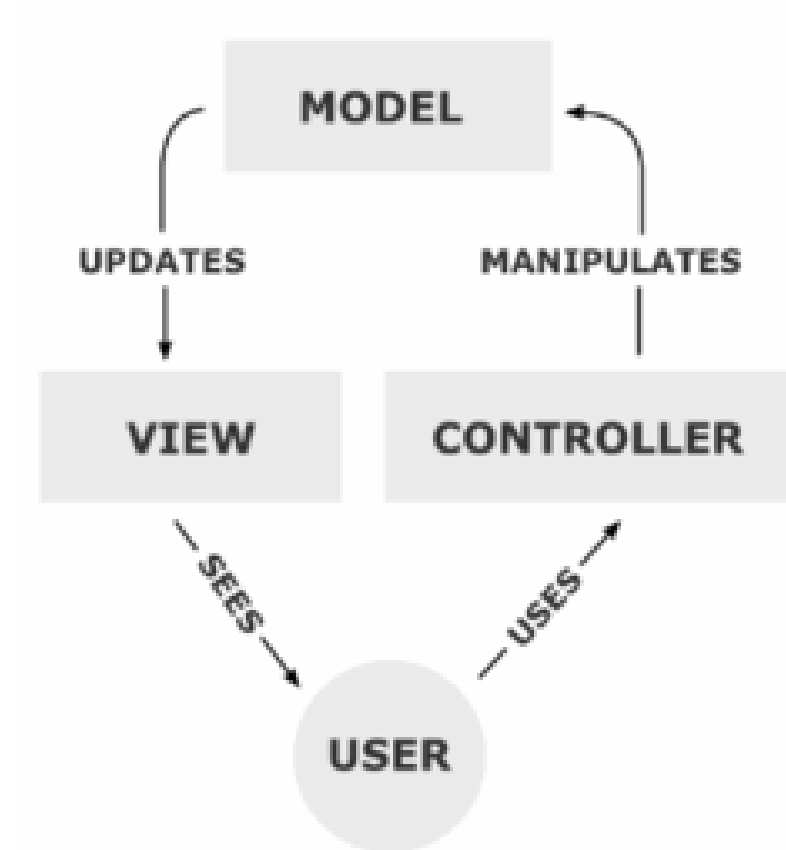


Рисунок 3.1 – Схема MVC

Структура бази даних програмної системи, що підлягає дослідженню, є аналогічною до типової структури бази даних інтернет-магазину загального або спеціалізованого призначення та містить у собі таблиці, що відповідають за зберігання даних про каталог магазину (категорії, товари, товарні пропозиції, фільтри, групи фільтрів), замовлення, клієнтів, SEO-дані, а також деякі допоміжні таблиці.

Схема частини бази даних, що відповідає за збереження даних про каталог, наведена на рисунку 3.2. Повну схему бази даних наведено у додатку А

Для візуалізації експерименту та поліпшення конфігурування системи було вирішено створити веб-інтерфейс, який повинен надавати змогу змінювати конфігурацію експериментів, структури бази даних та її безпосереднього змісту.

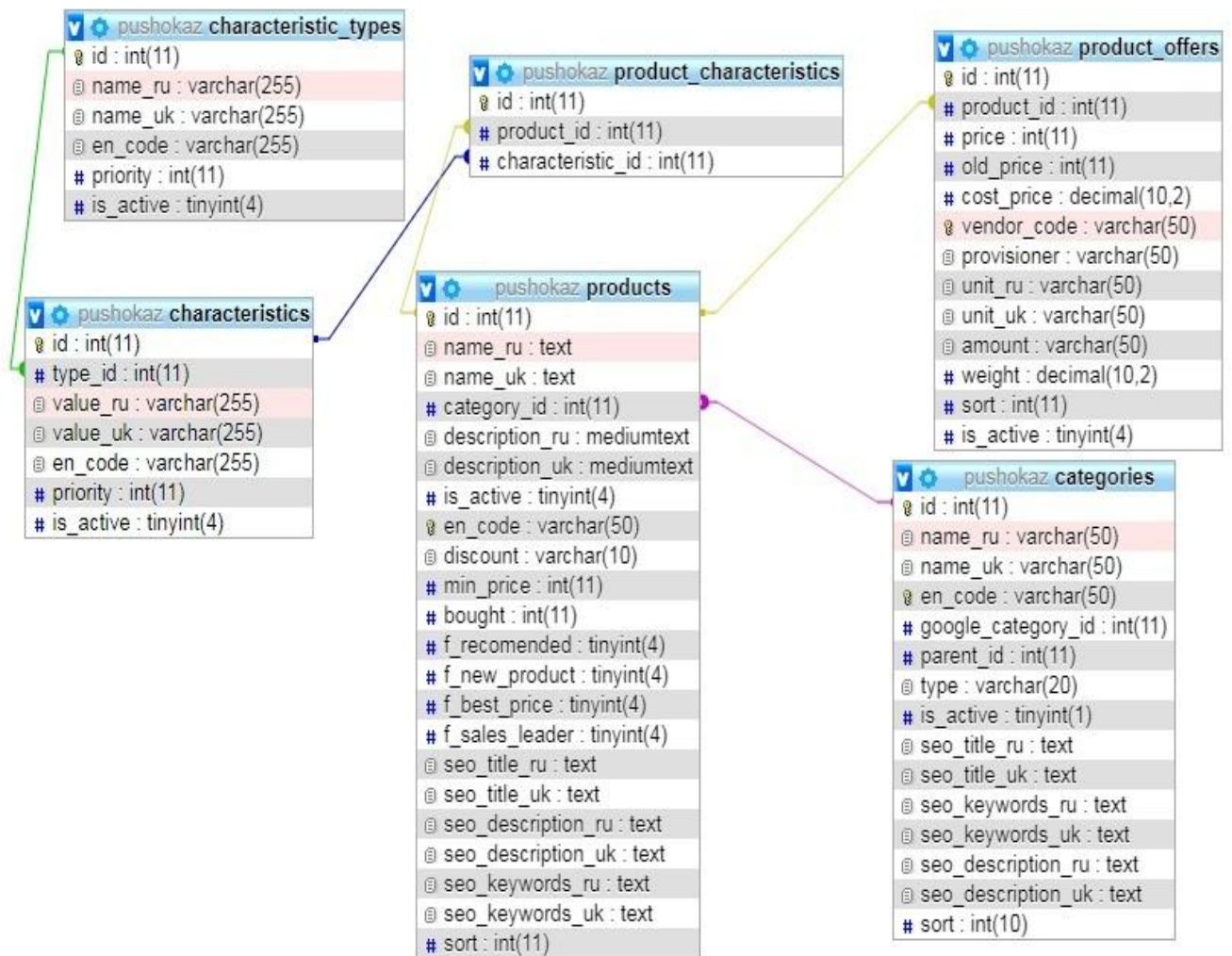


Рисунок 3.2 – Фрагмент схеми БД, що використовувався у дослідженні

Функції інтерфейсу було вирішено розділити за концептуальними ознаками та згрупувати по трьох окремих інструментальних блоках. Кожен з даних блоків відповідає за окремий аспект у дослідженні.

Зокрема блок інструментів для заповнення бази даних тестовими даними з відповідними характеристиками. Серед них: тип даних, середня довжина слова, середня кількість слів у назвах та кількість записів до бази даних. Вигляд даного блоку наведено на рисунку 3.3.

Для виконання заповнення експериментальної бази даних було розроблено набір характеристик даних, зокрема середня кількість слів у назві, середня кількість літер у слові, формат даних тощо, а також спроектовано та реалізовано окремий клас, за допомогою якого були сформовані тестові дані для заповнення таблиць.

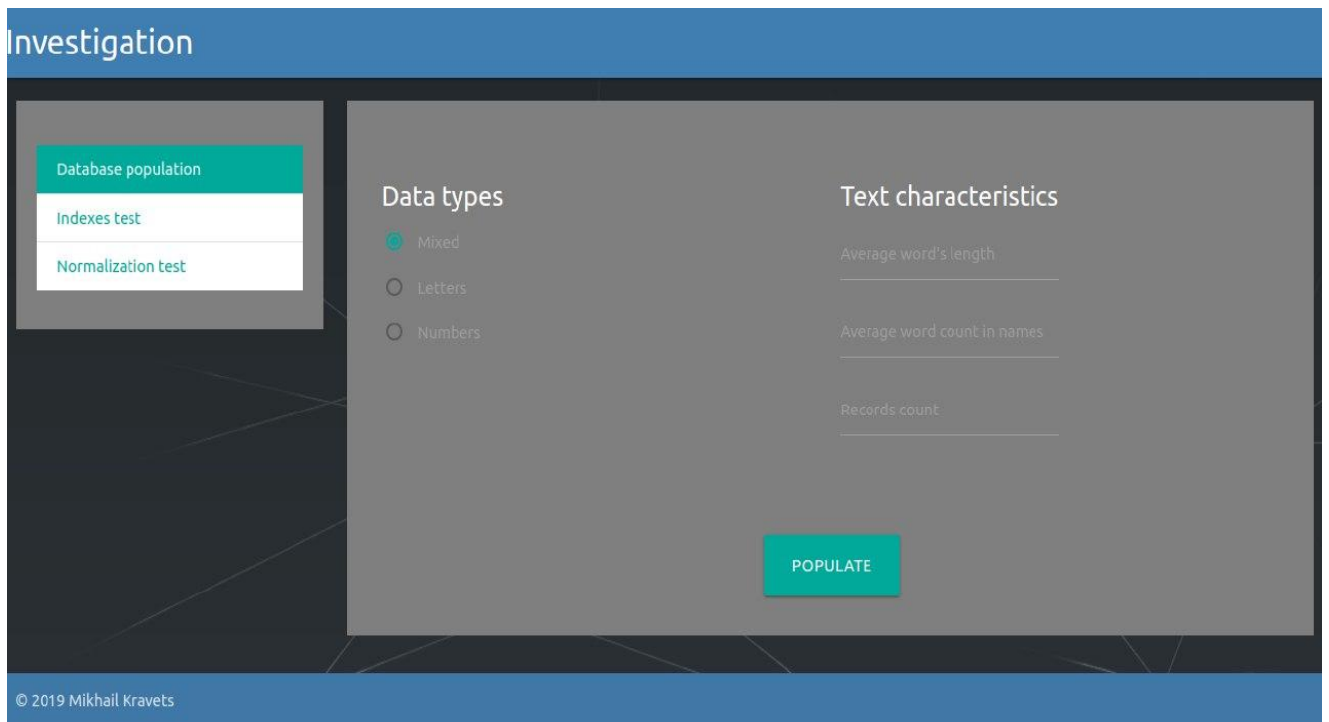


Рисунок 3.3 – Блок налаштувань заповнення БД

Блок налаштування індексації бази даних відповідає за додавання індексів на відповідних стовпчиках таблиць бази даних системи. Індокси додаються автоматично до стовпчиків, які можуть бути використаними при поєднанні таблиць за допомогою оператора JOIN. Даний блок наведено на рисунку 3.4.

На рисунку 3.5 наведено блок налаштування нормалізації бази даних, який відповідає за використання певного набору таблиць, які приведені до відповідної нормальної форми при зміні положення перемикача та натисканні на кнопку. Також наявна можливість вказування кількості запитів, що будуть виконуватися протягом експерименту. Блок з показниками часу змінюється після виконання експерименту та відображає отримані дані про мінімальний, максимальний та середній арифметичний час виконання запиту до бази даних.

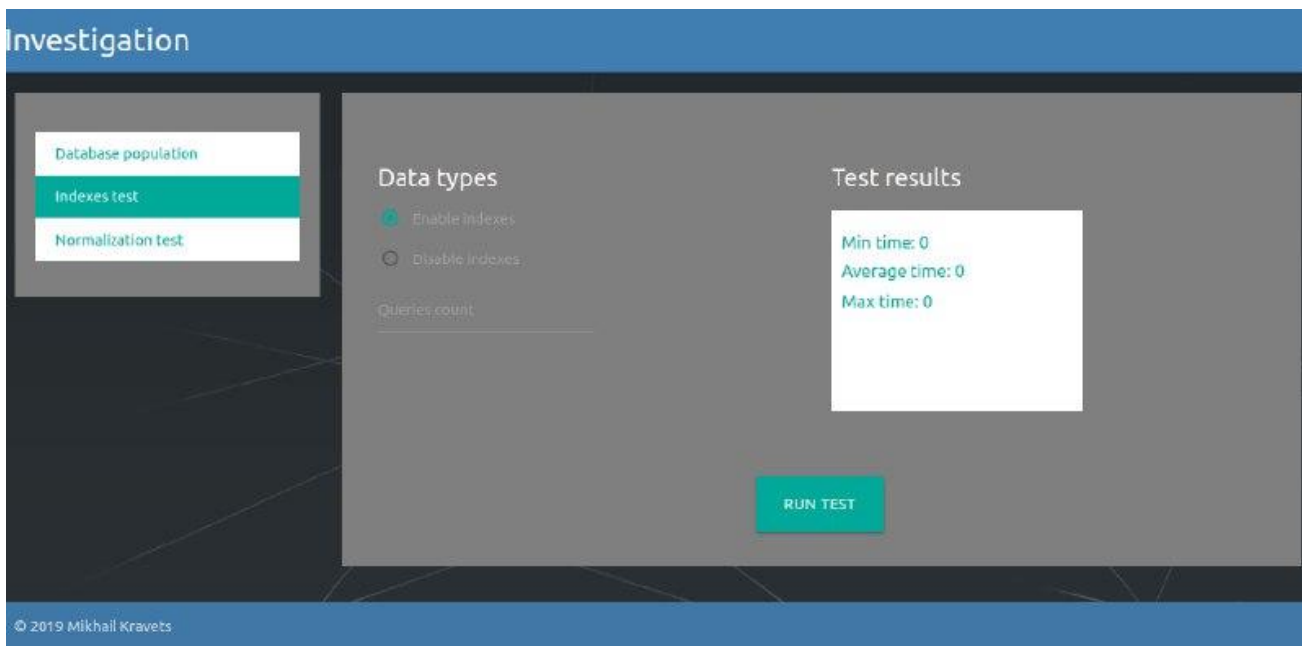


Рисунок 3.4 – Блок налаштування індексів у БД

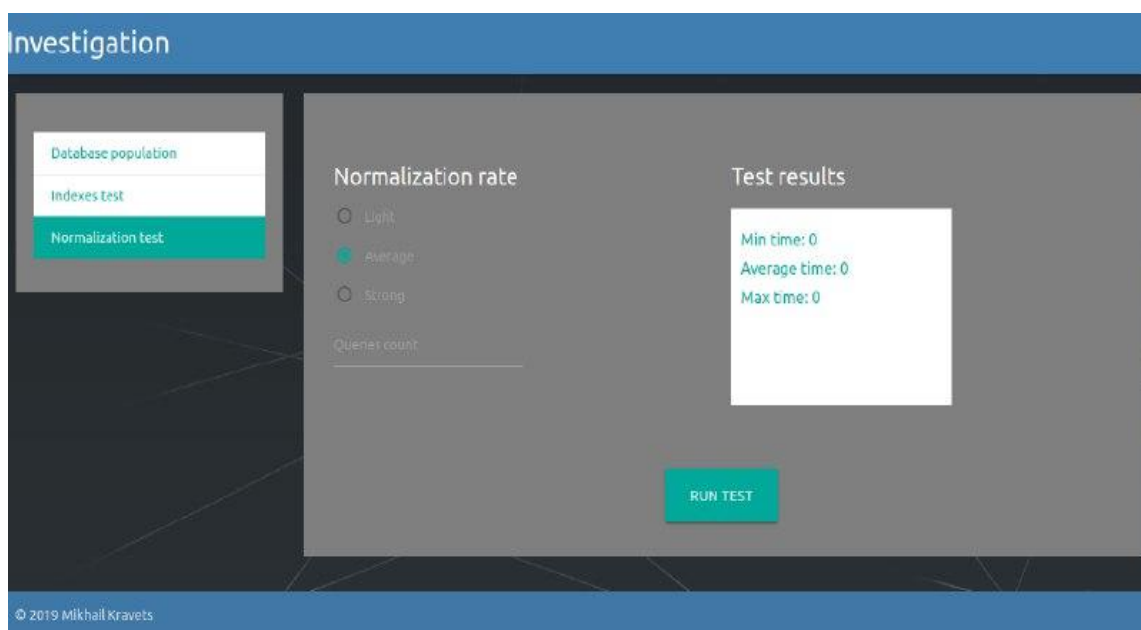


Рисунок 3.5 – Блок налаштування ступеню нормалізації БД

Реалізацію даного функціоналу також було виконано на рівні моделі шляхом здійснення запитів на зміну структури таблиці у базі даних.

Два останні інтерфейси використовувалися для проведення налаштування бази даних згідно запланованого експерименту, після чого виконувалося заповнення бази даних тестовими значеннями та відтворювався

набір запитів для отримання результату експерименту, а саме часу обробки запиту на рівні СКБД.

### 3.4 Опис проведених експериментів

Як було зазначено у попередньому розділі, для виведення закономірності у результатах дослідження було вирішено використовувати дисперсійний аналіз. Оскільки було обрано два фактори впливу на кінцевий час обробки SQL-запиту – у даному випадку треба використовувати двохфакторний дисперсійний аналіз. Проведення двохфакторного дисперсійного аналізу потребує виділення фіксованого набору значень незалежних змінних, а також проведення вичерпного набору експериментів з усіма комбінаціями значень обраних факторів.

У даному дослідженні буде використано два фактори впливу на час обробки запиту, а саме наявність індексів на стовпчиках таблиці, що беруть участь у фільтрації даних, або ж використовуються при поєднанні таблиць у запиті, а також ступінь нормалізації бази даних, а точніше її частини, що буде використана у експериментальних дослідженнях. У даному випадку будемо використовувати для цього фактору три значення:

- слабка нормалізація(еквівалентно 1 НФ);
- середня нормалізація(еквівалентно 2 НФ);
- сильна нормалізація(еквівалентно 3 НФ).

Вибір даного набору значень обумовлений тим, що найчастіше при нормалізації баз даних використовуються перші три нормальні форми.

Слід зазначити, що першим етапом проведення експериментів є заповнення бази даних тестовими значеннями. Основними характеристиками тестового набору даних були:

- тип даних(числові, текстові, змішані);
- середня кількість літер у слові;
- середня кількість слів у іменах;
- кількість товарів

Експериментальним шляхом було з'ясовано, що для забезпечення адекватності тестових даних, кількість категорій повинна орієнтовно дорівнювати 20 частині від кількості товарів, у той час, як кількість характеристик повинна дорівнювати орієнтовно 5 частині від кількості товарів, отже було вирішено використовувати дане співвідношення у якості базового. Кількість товарів було вирішено встановити на рівні 1000 одиниць, що приблизно дорівнює кількості товарів у інтернет-магазині середнього розміру. Процес заповнення бази даних програмно поділявся на три етапи:

- заповнення таблиць товарів та категорій;
- заповнення таблиць характеристик(фільтрів);
- заповнення асоціативної таблиці, що встановлює зв'язок типу M-N між сутностями товару та характеристики.

Наступним етапом експериментального дослідження після заповнення бази даних є проведення експериментів з усіма можливими наборами факторів впливу, що були описані у попередньому розділі. Загальна кількість таких комбінацій дорівнювала шести через те, що кількість станів для факторів впливу дорівнювала 2 та 3 відповідно. Результатом кожного експерименту слугував набір значень часу обробки запиту.

Для забезпечення точності результатів експерименту та адекватності дисперсійного аналізу, кількість запитів до бази даних у кожному з експериментів була однаковою та дорівнювала 1000 запитів. У серверному коді було впроваджено можливість експорту отриманих даних з кожного експерименту у файл .csv, що дозволило підрахувати середнє значення для часу обробки запиту у секундах для подальшого використання у дисперсійному аналізі. Дану таблицю приведено на рисунку 3.6.

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

Рисунок 3.6 – Результат експериментів

Позначимо фактори впливу(ступінь нормалізації та наявність індексів) як А та В відповідно, тоді окремі значення факторів позначимо як А1,А2,А3 та В1,В2 відповідно. Усі середні значення часу для кожного експерименту зводимо до таблиці Excel.

## 4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ

### 4.1 Проведення дисперсійного аналізу

Як було зазначено у попередніх розділах, для аналізу отриманих експериментальних даних було обрано багатофакторний дисперсійний аналіз, а оскільки у даному випадку факторів впливу розглядається два – будемо використовувати двохфакторний дисперсійний аналіз. У якості нульової гіпотези візьмемо гіпотезу, що всі використані фактори впливу не впливають на швидкість обробки запиту на рівні СКБД.

Для проведення дисперсійного аналізу було прийняте рішення використати вбудований модуль MS Excel для виконання аналізу даних. Для використання даного модулю потрібно скористуватись меню "Аналіз даних" у вкладці "Дані". Вигляд меню "Аналіз даних" приведено на рисунку 4.1

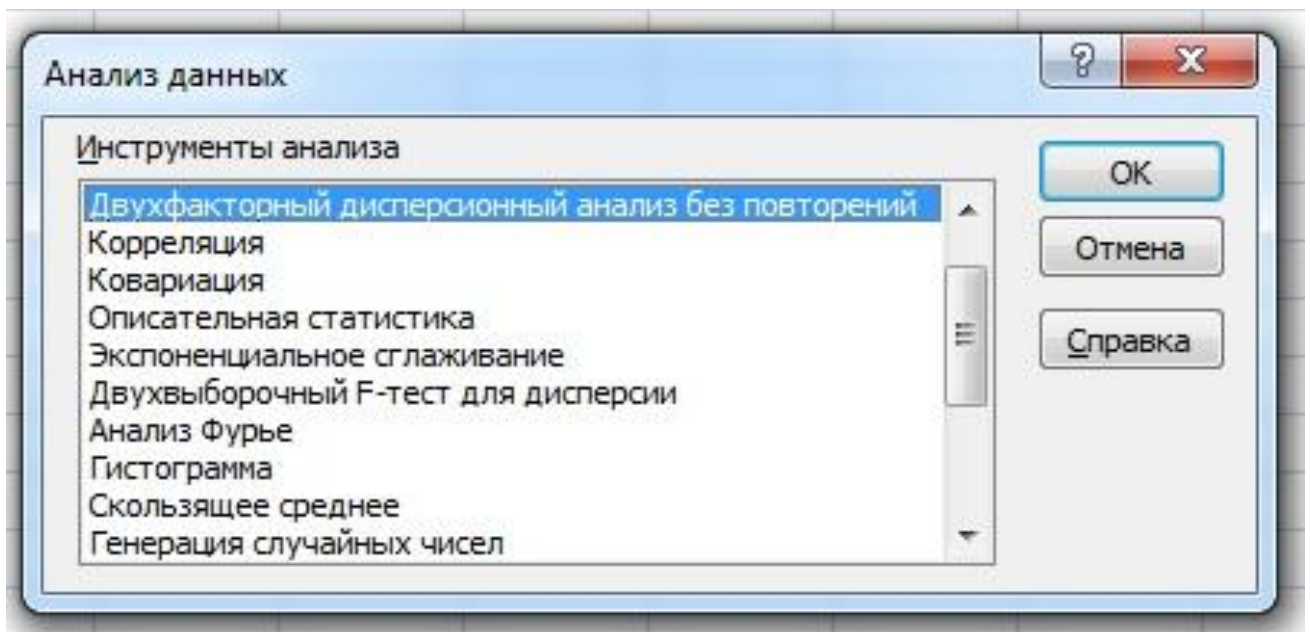


Рисунок 4.1 – Меню аналізу даних у MS Excel

У меню інструментів аналізу даних обираємо пункт "Двохфакторний дисперсійний аналіз без повторень" та виділяємо інтервал вхідних значень. Також вказуємо множину "Альфа"(рівень значимості), яка по суті є показником точності отриманих експериментальних даних. Встановлюємо

значення 0.28 для Альфа, що буде означати допущення максимальної похибки у 28%, що є досить низьким рівнем похибки.

Рівень похибки у 28% показує, що обрана модель є досить точною у описанні даної системи, що потребує оптимізації. Меню налаштувань дисперсійного аналізу наведено на рисунку 4.2

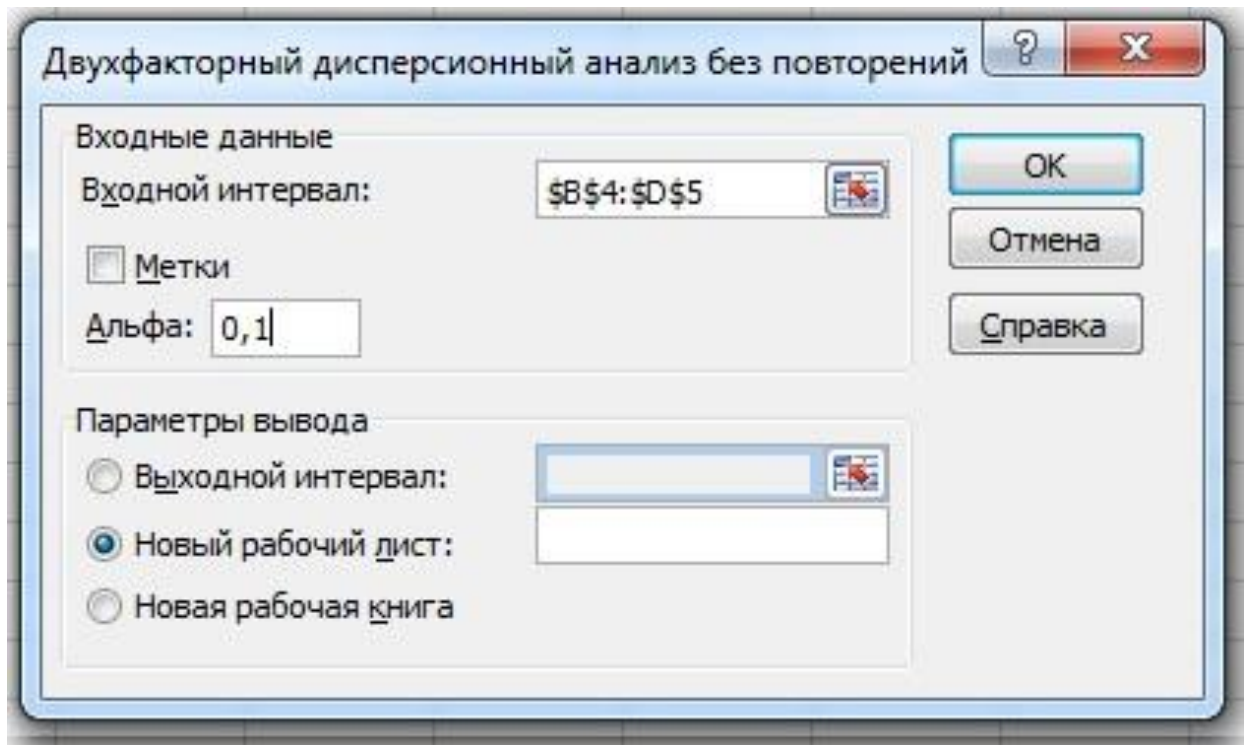


Рисунок 4.2 – Вікно налаштувань дисперсійного аналізу

Після запуску дисперсійного аналізу, MS Excel формує таблиці з обчисленими значеннями дисперсії та інших показників. Дані показники приведені у таблиці 4.1.

У даній таблиці можна побачити, що MS Excel виконує обчислення дисперсії окремо для рядків та стовпчиків таблиці вхідних даних, після чого обчислює показник дисперсії для кожного з них.

У таблиці приведено суму за кожним рядком та стовпчиком початкових даних, їх середнє значення та показник дисперсії.

Таблиця 4.1 – Результат дисперсійного аналізу

Результати	Рахунок	Сума	Середнє	Дисперсія
Рядок 1	3	0,0004925	0,000164167	1,79083E-10
Рядок 2	3	0,00043	0,000143333	5,62333E-10
Стовпчик 1	2	0,000327	0,0001635	4,5E-12
Стовпчик 2	2	0,00027	0,000135	5,78E-10
Стовпчик 3	2	0,0003255	0,00016275	4,96125E-10

Дана таблиця не є кінцевим результатом дисперсійного аналізу. MS Excel також обчислює критерій Фішера та Р-значення, які саме і є найважливішими показниками дисперсійного аналізу. Також у таблиці приведено середнє квадратичне значення. Результат обчислень критерію Фішера, Р-значень та ступенів свободи за рядками та стовпчиками приведений у таблиці 4.2, що також була сформована за допомогою обчислень у MS Excel.

Таблиця 4.2 – Результат дисперсійного аналізу

Джерело варіації	SS	df	F	Р-Значення	F критичне
Рядки(B)	6,51042E-10	1	3,045215358	0,223092791	2,033071
Стовпці(A)	1,05525E-09	2	2,467939973	0,288355625	2,44827
Похибка	4,27583E-10	2			
Разом	2,13388E-09	5			

У даній таблиці представлені обчислені значення для критерію Фішера та Р-значення.

#### 4.2 Аналіз результатів експерименту

З таблиці 4.2 видно, що величина  $F$  є меншою, ніж  $F$  критичне. У даному випадку це означає, що нульову гіпотезу відносно факторів впливу, спростовано та вони дійсно мають значний вплив на швидкість обробки запитів до бази даних.

$P$ -значення показує ймовірність того, що фактор, для якого обчислено  $P$ -значення буде впливати на результат експерименту, іншими словами у даному випадку ступінь нормалізації бази даних впливатиме на результат експерименту з ймовірністю приблизно 0.28, у той же час, як наявність індексів на необхідних стовпчиках таблиць бази даних впливатиме на кінцевий результат з ймовірністю 0.22, отже виходячи з результатів експериментів, що були проведені, ступінь нормалізації впливатиме на швидкість обробки запиту значніше, ніж наявність індексів при умові збігу порядку кількості даних з експериментальними та використання подібної до експериментальної структури бази даних.

#### 4.3 Практичне застосування результатів дослідження

Як було зазначено вище, обидва обраних фактори з великою ймовірністю значно впливають на швидкість обробки SQL-запиту, але ж практична користь даного дослідження полягає у іншому. У таблиці 4.3 приведено результати статистичного дослідження комбінацій факторів.

Для отримання наочного вигляду результатів, за допомогою MS Excel побудуємо гістограму результатів з факторами А та В по горизонталі і часом по вертикалі. Гістограму приведено на рисунку 4.3

Таблиця 4.3 – Результати експериментів

Фактори	A1	A2	A3
B1	0,000162	0,000152	0,0001785
B2	0,000165	0,000118	0,000147

З діаграми виходить, що найоптимальніший показник часу був відмічений при 2НФ, тобто при певній денормалізації та при наявності індексів. Це свідчить про те, що дана комбінація факторів оптимізації у умовах, що є подібними до експериментальних, будуть найоптимальнішим рішенням для прискорення обробки SQL-запиту на рівні СКБД. Цей результат може бути корисним при використанні у процесі розробки систем з подібною структурою.

Досить показовим є результат, що стосується 1НФ. Час обробки запитів при наявності індексів виявився вище, ніж за їх відсутності. Такий результат може виникати через особливість архітектури індексів.

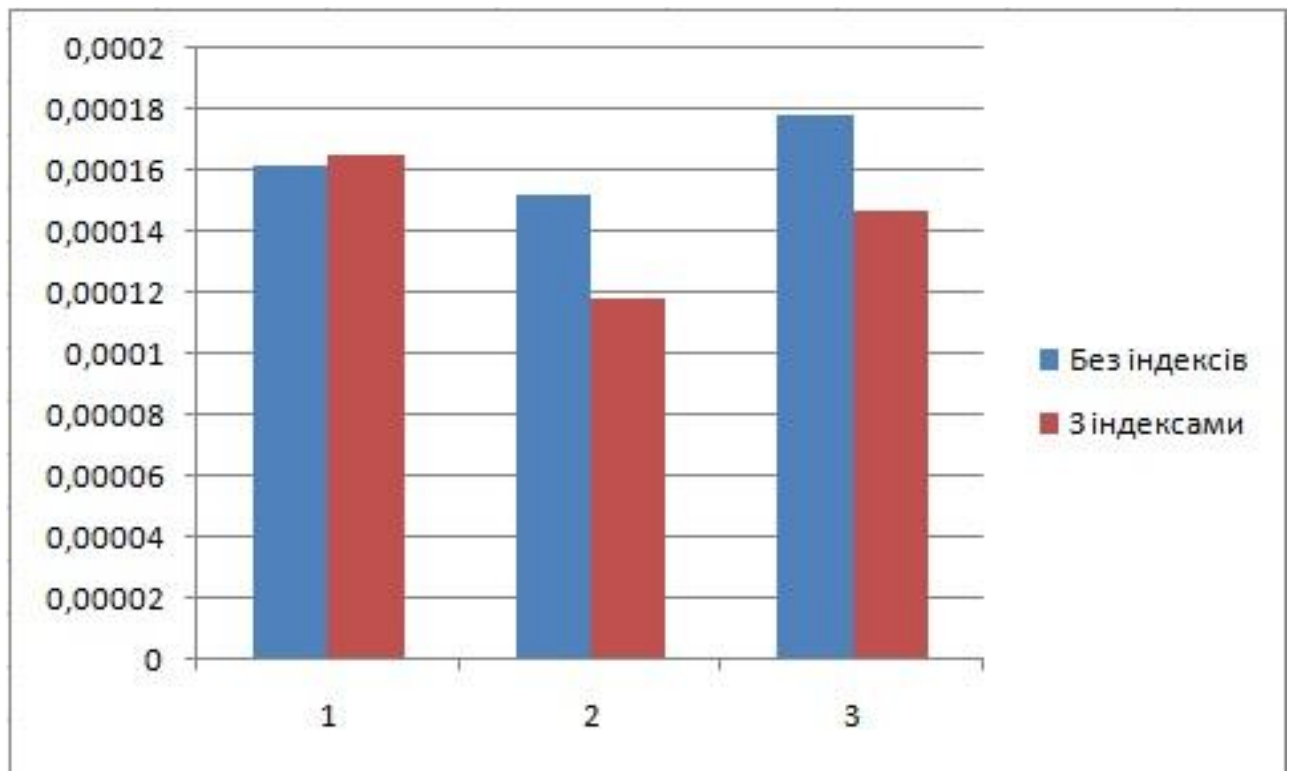


Рисунок 4.3 – Гістограма з результатами експериментів

Індекс являє собою таблицю значень та найкраще за все працює на наборі унікальних значень, що є досить складно. умовою у разі використання таблиць, що приведені до першої нормальної форми та через це можуть мати значну кількість дублікатів даних у рядках, тому у таких випадках індекс, що зазвичай використовується для прискорення обробки запитів, значно

сповільнює її. Це відбувається з причини того, що СКБД спочатку зчитує індекс, а вже після цього, коли не знаходить у ньому унікального значення, починає виконувати прямий пошук значення по таблиці, а через наявність у індексі значної кількості дублікатів, зчитування індексу займає значно більше часу, а також не дає змоги знайти за індексом унікальне значення, що саме і призводить до отриманого результату.

Отже, даний результат демонструє, що використання індексів іноді може бути не доцільним, а надмірна нормалізація бази даних при деяких обставинах може бути причиною сповільнення взаємодії з нею.

## ВИСНОВКИ

Під час проведення дослідження були проаналізовані головні способи та етапи математичного моделювання системи, а саме реляційної бази даних. Найбільш відповідною математичною моделлю у даному випадку виявилась математична модель статистичного типу, яка передбачає результуючий час завдяки наявності великої кількості статистичних даних.

Було детально досліджено вплив ступеню нормалізації бази даних та наявність індексів на швидкість обробки запиту на стороні СКБД, а також проведено двохфакторний дисперсійний аналіз для підтвердження результатів статистичного дослідження, у результаті якого виявилось, що обидва фактори є достатньо вагомими, щоб значно впливати на швидкість обробки SQL-запиту.

Також було досліджено вплив перерахованих факторів при використанні їх на великих наборах змішаних даних, що зазвичай застосовуються.

У дослідженні використовувалася база даних інтернет-магазину типової структури, що дає змогу спроектувати результати даного дослідження на реальні проекти та досягти приросту у швидкості взаємодії з базою даних.

Окрім цього, результати даного дослідження можуть бути використані у якості інструкції при проектуванні нових баз даних реляційного типу, що матимуть схожий набір характеристик.

У подальшому необхідно ускладнити задачу дослідження шляхом додавання більшої кількості факторів впливу на швидкість обробки SQL-запиту, а також дослідити закономірність ступеня впливу факторів у залежності від розміру тестового набору даних, що використовується.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Шварц, Б., MySQL. Оптимизация производительности, 2-е изд. Москва: Символ, 2010. - 57 с
2. Колесников О. В. Основи наукових досліджень. 2-ге вид. випр.. та доп. Навч. посіб.- К.: Центр учбової літератури, 2011. - 144 с.
3. Кузнецов, А. А., Дисперсионный анализ сетевого трафика для обнаружения и предотвращения вторжений в телекоммуникационных системах и сетях. 2014. - 227 с.
4. Дивак, М. П. Задачі математичного моделювання статистичних систем з інтервальними даними. 2011. - 134 с.
5. Сайт публікацій матеріалів для навчання. Електронний ресурс, URL: <https://lektsii.org/> (дата звернення: 10.11.2019).
6. Офіційний сайт MySQL. - URL: <http://mysql.com> (дата звернення: 09.11.2019)
7. Єрохін, А. Л.; Самсонов, В. В. Методи та засоби інтернет–технологій: навч. посіб. Харків: компанія СМІТ, 2008. - 264 с.
8. Letkowski, Jerzy. Doing database design with MySQL. Journal of Technology Research, 2015, - 134 с.
9. Березький, О. М., Розроблення реляційної бази даних інтелектуальної системи автоматизованої мікроскопії. Науковий вісник НЛТУ України, 2017.
10. Vanier, Eric; Shah, Birju; Malepati, Tejaswi. Advanced MySQL 8, 2019.
11. Bellatreche, Ladjel; Valduriez, Patrick; Morzy, Tadeusz. Advances in databases and information systems., 2018, - 286 с.
12. WANG, Xing,. Database optimization concepts in fast response environments. U.S. Patent Application No 15/592,709, 2017.
13. Мирошник, С. Н.; Гончар, Д. Р.; Фуругян, М. Г. Оптимизация структуры базы данных реального времени. Управление большими системами: сборник трудов, 2017, 66