Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: Дослідження програмних архітектур SQL та NoSQL баз даних для потреб "розумних міст"

Виконав: студент VI курсу, групи СНм-61
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

_____ Онуферко В.А.
(підпис) (прізвище та ініціали)

Керівник _____ Матійчук Л.П.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ Дуда О.М.
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ Боднарчук І.О.
(підпис) (прізвище та ініціали)

Рецензент _____ Михалик Д.М.
(підпис) (прізвище та ініціали)

Тернопіль
2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
«_26_» _грудня_ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Онуферко Віталію Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження програмних архітектур SQL та NoSQL баз даних для потреб "розумних міст"

Керівник роботи Матійчук Любомир Павлович, к.е.н., доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «_24_» _листопада_ 2023 року № _4/7-1099_

2. Термін подання студентом завершеної роботи 27 грудня 2023р.

3. Вихідні дані до роботи Наукові публікації про програмні архітектури, SQL, NoSQL, бази даних та «розумні міста».

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1 Стан предметної області використання SQL та NoSQL баз даних у «розумних містах» та організація процесів наукометричного пошуку. 2 Аналіз результатів наукометричного пошуку щодо використання SQL та NoSQL баз даних у «розумних містах». 3. Дослідження особливостей використання різних SQL і NoSQL базах даних «розумних міст». 4 Охорона праці та безпека в надзвичайних ситуаціях. Висновки. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1 Титульна сторінка. 2 Тема, Мета, Об'єкт, Предмет дослідження. 3 Завдання дослідження. 4 Актуальність дослідження. 5 SQL та NOSQL баз даних у «розумних містах». 6 Основні функції NoSQL. 7. Комбінації пошукових запитів. 8. Категоризація вибраних наукових публікацій. 9. Бази даних, використані в окремих дослідженнях. 10. Структури зберігання баз даних SQL і NoSQL «розумних міст». 11. Характеристики MongoDB і Oracle RDBMS. 12. Геометричні об'єкти, що підтримуються базами даних SQL і NoSQL. 13. Висновки. 17. Завершальний слайд.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	25.11.2023	*Виконано*
2.	Підбір наукових джерел про програмні архітектури,	26.11.2023-28.11.2023	*Виконано*
	SQL, NoSQL, бази даних та «розумні міста».		
3.	Опрацювання наукових публікацій та збір даних по	29.11.2023-1.12.2023	*Виконано*
	темі роботи		
4.	Виконання дослідження згідно мети кваліфікаційної	2.12.2023-4.12.2023	*Виконано*
	роботи		
5.	Оформлення розділу «Стан предметної області	5.12.2023-7.12.2023	*Виконано*
	використання SQL та NoSQL баз даних у «розумних		
	містах» та організація процесів наукометричного		
	пошуку»		
6.	Оформлення розділу «Аналіз результатів	8.12.2023-10.12.2023	*Виконано*
	наукометричного пошуку щодо використання SQL та		
	NoSQL баз даних у «розумних містах»»		
7.	Оформлення розділу «Дослідження особливостей	11.12.2023-13.12.2023	*Виконано*
	використання різних SQL і NoSQL базах даних		
	«розумних міст»»		
8.	Виконання завдання до підрозділу «Охорона праці»	14.12.2023-15.12.2023	*Виконано*
9.	Виконання завдання до підрозділу «Безпека в	16.12.2023-17.12.2023	*Виконано*
	надзвичайних ситуаціях»		
10.	Оформлення кваліфікаційної роботи	18.12.2023-19.12.2023	*Виконано*
11.	Нормоконтроль	19.12.2023-20.12.2023	*Виконано*
12.	Перевірка на плагіат	21.12.2023	*Виконано*
13.	Попередній захист кваліфікаційної роботи	22.12.2023	*Виконано*
14.	Захист кваліфікаційної роботи	27.12.2023	

АНОТАЦІЯ

Дослідження програмних архітектур SQL та NoSQL баз даних для потреб «розумних міст» // Кваліфікаційна робота освітнього рівня «Магістр» // Онуферко Віталій Андрійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2023 // С. 75, рис. – 13, табл. – 6, кресл. – , додат. – 1, бібліогр. – 93.

Ключові слова: агрегація, бази даних, великі дані, порівняння, розумне місто, MapReduce, NoSQL, SQL.

Кваліфікаційна робота присв'ячена розробці дослідженню програмних архітектур SQL та NoSQL баз даних для потреб «розумних міст». В першому розділі кваліфікаційної роботи освітнього рівня «Магістр» висвітлено актуальність дослідження предметної області використання SQL та NoSQL баз даних у «розумних містах». Описано стан проблемної області використання SQL та NoSQL баз даних для «розумних міст». Подано критерії пошуку наукових публікацій щодо використання SQL та NoSQL баз даних у«розумних містах». Розглянуто задіяні пошукові ресурси. Сформовано стратегію наукометричного пошуку матеріалів щодо використання SQL та NoSQL баз даних для «розумних міст». Описано процес і критерії відбору наукових публікацій. В другому розділі кваліфікаційної роботи проведено аналіз і класифікацію результатів наукометричного пошуку. Подано результати пошуку наукових публікацій. Проаналізовано результати наукометричного пошуку. Виконано моделювання даних NoSQL MongoDB для потреб «розумних міст». В третьому розділі кваліфікаційної роботи описано ефективність DBaaS, взаємодія та переносимість даних у різних NoSQL базах даних «розумних міст». Проведено обговорення та класифікація особливостей використання різнотипових баз даних в проектах класу «розумне місто». Висвітлено прогалини між різнотиповими базами даних «розумних міст». Подано результати дослідження програмних архітектур SQL та NoSQL баз даних для потреб «розумних міст».

ANNOTATION

Software architectures research on SQL and NoSQL Database for "Smart cities" // The educational level "Master" qualification work // Onuferko Vitalii Andriiovych // Ternopil Ivan Pulyuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, SNm-61 group // Ternopil, 2023 // P. 75, fig. – 13, tables – 6, posters – , annexes – 1, ref. – 93.

Key words: aggregation, databases, big data, comparison, smart city, MapReduce, NoSQL, SQL.

The qualification work is devoted to the development and research of software architectures of SQL and NoSQL databases for the needs of "smart cities". In the first section of the qualifying work of the "Master's" educational level, the relevance of research in the subject area of the use of SQL and NoSQL databases in "smart cities" is highlighted. The state of the problem area of using SQL and NoSQL databases for "smart cities" is described. The search criteria for scientific publications on the use of SQL and NoSQL databases in "smart cities" are provided. The search resources involved are considered. A strategy for scientometric search for materials on the use of SQL and NoSQL databases for "smart cities" was formed. The process and criteria for selecting scientific publications are described. In the second section of the qualification work, the analysis and classification of the results of the scientometric search was carried out. The results of the search for scientific publications are presented. The results of the scientometric search were analyzed. Performed modeling of NoSQL MongoDB data for the needs of "smart cities". The third chapter of the qualification paper describes the effectiveness of DBaaS, interoperability and data portability in various NoSQL databases of "smart cities". The discussion and classification of the features of the use of various types of databases in projects of the "smart city" class was carried out. Gaps between various databases of "smart cities" are highlighted. The results of the study of software architectures of SQL and NoSQL databases for the needs of "smart cities" are presented.

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

ACID (англ. Atomicity, Consistency, Isolation, Durability) – Доступність, послідовність, ізоляція та довговічність.

CSP (англ. Cloud Service Provider) – Постачальник хмарних послуг.

SLR (англ. System of Location Resources) – Система розміщення ресурсів.

OLTP (англ. Online Transaction Processing) – Онлайнова обробка транзакцій.

RDBMS (англ. Relational Database Management System) – Реляційна система керування базами даних.

ЗМІСТ

ВСТУП 8
1 СТАН ПРЕДМЕТНОЇ ОБЛАСТІ ВИКОРИСТАННЯ SQL ТА NOSQL БАЗ ДАНИХ У «РОЗУМНИХ МІСТАХ» ТА ОРГАНІЗАЦІЯ ПРОЦЕСІВ НАУКОМЕТРИЧНОГО ПОШУКУ 11
1.1 Актуальність дослідження предметної області використання SQL та NoSQL баз даних у «розумних містах» 11
1.2 Стан проблемної області використання SQL та NoSQL баз даних для «розумних міст» 15
1.3 Критерії пошуку наукових публікацій щодо використання SQL та NoSQL баз даних у«розумних містах» 16
1.4 Задіяні пошукові ресурси 17
1.5 Стратегія наукометричного пошуку матеріалів щодо використання SQL та NoSQL баз даних для «розумних міст» 18
1.6 Процес і критерії відбору наукових публікацій 19
1.7 Критерії включення наукових публікацій 20
1.8 Критерії виключення наукових публікацій 21
1.9 Збір і вилучення даних щодо наукових публікацій 21
1.10 Висновок до першого розділу 22
2 АНАЛІЗ РЕЗУЛЬТАТІВ НАУКОМЕТРИЧНОГО ПОШУКУ ЩОДО ВИКОРИСТАННЯ SQL ТА NOSQL БАЗ ДАНИХ У «РОЗУМНИХ МІСТАХ» 23
2.1 Аналіз і класифікація результатів наукометричного пошуку 23
2.2 Результати пошуку наукових публікацій 23
2.3 Аналіз результатів наукометричного пошуку 27
2.4 Моделювання даних NoSQL MongoDB для потреб «розумних міст» 37
2.5 Висновок до другого розділу 39
3 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ВИКОРИСТАННЯ РІЗНИХ SQL І NOSQL БАЗАХ ДАНИХ «РОЗУМНИХ МІСТ» 40

3.1 Ефективність DBaaS, взаємодія та переносимість даних у різних NoSQL базах даних «розумних міст» 40
3.2 Обговорення та класифікація особливостей використання різнотипових баз даних в проектах класу «розумне місто» 44
3.3 Прогалини між різнотиповими базами даних «розумних міст» 50
3.4 Результати дослідження програмних архітектур SQL та NoSQL баз даних для потреб «розумних міст» 53
3.5 Висновок до третього розділу 54
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ 55
4.1 Долікарська допомога при контузіях 55
4.2 Фактори що впливають на функціональний стан користувача комп'ю тера 57
4.3 Висновок до четвертого розділу 63
ВИСНОВКИ 64
ПЕРЕЛІК ДЖЕРЕЛ 66
ДОДАТКИ

ВСТУП

Актуальність теми. В контексті «розумних міст» грамотна архітектура програмного забезпечення для баз даних SQL і NoSQL є критично важливою для ефективного управління великими обсягами даних, які генеруються пристроями Інтернету речей (IoT). Бази даних SQL є традиційним вибором для зберігання і керування структурованими даними, такими як транзакції, облікові записи та інші дані, які можна легко визначити та класифікувати. Вони добре підходять для завдань обробки онлайнових транзакцій (OLTP), таких як обробка замовлень або бронювання. Бази даних NoSQL, з іншого боку, краще підходять для зберігання і керування неструктурованими даними, такими як дані з давачів, відео та аудіо. Вони можуть масштабуватися горизонтально, що робить їх ідеальними для зберігання великих обсягів даних.

Організації, які розробляють розумні міста, повинні вибрати парадигму бази даних, яка відповідає їхнім конкретним потребам. Якщо вони потребують бази даних для зберігання і керування структурованими даними, такими як дані про транспорт або комунальні послуги, база даних SQL може бути хорошим вибором. Якщо вони потребують бази даних для зберігання і керування неструктурованими даними, такими як дані з давачів або відео, база даних NoSQL може бути кращим вибором. Переносимість та сумісність даних є ще однією важливою проблемою для організацій, які використовують хмарні служби для зберігання даних. Різні хмарні платформи (IaaS, PaaS, SaaS і DBaaS) використовують різні парадигми бази даних. Це може ускладнити перенесення даних між різними платформами.

Дослідження показують, що бази даних NoSQL можуть бути найкращим вибором для аналізу великих даних у розумних містах. Вони добре підходять для зберігання і керування великими обсягами неструктурованих даних, які генеруються пристроями IoT. Це дані, які можна використовувати для аналізу тенденцій, виявлення аномалій та прийняття рішень.

Ось кілька конкретних прикладів того, як бази даних SQL і NoSQL можуть використовуватися в розумних містах:

– Бази даних SQL можна використовувати для зберігання і керування транзакційними даними, такими як платежі за паркування або квитки на громадський транспорт. Це може допомогти покращити ефективність та прозорість цих систем.

– Бази даних NoSQL можна використовувати для зберігання і керування даними з давачів, такими як дані про температуру, рівень забруднення або дорожній рух. Це дані можна використовувати для аналізу тенденцій, виявлення аномалій та прийняття рішень про управління містом.

– Бази даних NoSQL також можна використовувати для зберігання і керування відео та аудіо даними. Це дані можна використовувати для моніторингу громадської безпеки, виявлення злочинів та забезпечення безпеки громадян.

Очікується, що використання баз даних SQL і NoSQL у розумних містах буде зростати в майбутньому. Це пов’язано з тим, що технологія IoT продовжує розвиватися, а також із зростаючим попитом на розумні рішення для міст. Тому дослідження програмних архітектур SQL та NoSQL баз даних для потреб «розумних міст» є актуальним напрямком досліджень.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Магістр» є підвищення рівня повноти подання інформації щодо програмних архітектур SQL та NoSQL баз даних та їх особливостей використання в інноваційних проектах «розумних міст». Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

– Проаналізувати існуючі підходи і методи документації SQL і NoSQL, враховуючи обробку великих даних «розумних міст».

– Виконати систематичний огляд літератури, пов’язаної з базами даних SQL і NoSQL для потреб «розумних міст».

– Оцінити сильні та слабкі сторони баз даних SQL і NoSQL для потреб «розумних міст».

Об’єкт дослідження: оцінка придатності програмних архітектур SQL та NoSQL баз даних для задоволення вимог до зберігання та обробки даних у розумних містах.

Предмет дослідження. Програмні архітектури SQL та NoSQL баз даних та їх застосування у розумних містах.

Наукова новизна одержаних результатів кваліфікаційної роботи полягає у тому, що отримав оригінальне прикладне використання метод наукометричного пошуку снігової кульки.

Практичне значення одержаних результатів. Сформовано рекомендаційні описи програмних архітектур SQL та NoSQL баз даних для потреб «розумних міст».

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на XI науково-технічної конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя (м. Тернопіль, 2023 р.).

Публікації. Основні результати кваліфікаційної роботи опубліковано у двох працях конференції (Див. додатки А).

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури з 93 найменувань та одного додатка. Загальний обсяг кваліфікаційної роботи складає 75 сторінки, з них 45 сторінок основного тексту, який містить 13 рисунків та 6 таблиць.

1 СТАН ПРЕДМЕТНОЇ ОБЛАСТІ ВИКОРИСТАННЯ SQL ТА NOSQL БАЗ ДАНИХ У «РОЗУМНИХ МІСТАХ» ТА ОРГАНІЗАЦІЯ ПРОЦЕСІВ НАУКОМЕТРИЧНОГО ПОШУКУ

1.1 Актуальність дослідження предметної області використання SQL та NoSQL баз даних у «розумних містах»

Архітектура конкретного програмного забезпечення в проектах класу «розумне місто» враховує нефункціональні характеристики:

– надійність;
– зручність використання;
– масштабованість;
– продуктивність;
– сумісність;
– портативність;
– адаптивність;
– шардинг даних.

Між набором атрибутів якості завжди є компроміси, і архітектор програмного забезпечення стикається з важким завданням їх балансування. Системи великих даних «розумних міст» [1] є внутрішньо розподіленими. Труднощі з доступністю та узгодженістю даних виникають через розподіл і реплікацію даних у великих міських системах даних. У зв’язку зі збільшенням кількості застосуків даних технології баз даних зазнали значних змін. Впродовж більш ніж десятиліття бази даних NoSQL зросли експоненційно, хоча класична автоматизація баз даних збереглася. Традиційний макет передбачає жорстку структуру схеми, що призводить до неясності масштабування та перешкоджає модифікації даних у кластерах. Навпаки, бази даних NoSQL підтримують прості прототипи. Основні властивості проектів баз даних NoSQL включають:

– Безсхемна структура.
– Дає змогуефективне та динамічне зростання представлень даних.

– Горизонтальне масштабування за допомогою колекцій реплікації даних і шардингу над масивними кластерами.

За останні роки численні муніципальні організації «розумних міст» накопичили величезні обсяги даних [2], які реляційні бази даних не можуть ефективно обробити. За останні чотири десятиліття відбулося величезне зростання використання реляційних баз даних. Вони відповідають атрибуту ACID – доступність, послідовність, ізоляція та довговічність, і призначені для структурованих даних. Хоча «великі дані» включають інструменти та технології, які керують величезними обсягами даних у будь-якому масштабі, ці інструменти та технології є масштабованими в контексті їх використання для потреб «розумних міст». Великі дані включають 5V:

– обсяг;
– швидкість;
– різноманітність;
– правдивість;
– цінність.

Водночас вони включають величезну кількість неструктурованих даних «розумних міст» різного характеру. Численні фреймворки, включаючи Hadoop/MapReduce, Spark, Flink і Samza, використовуються для обробки великих даних [1].

Тема продуктивності та оптимізації запитів SQL для міських корпоративних, виробничих, паралельних баз даних і великих даних [3] привернула підвищену увагу в останні роки. Неефективні та неоптимізовані запити можуть споживати ресурси системи та сервера, що призведе до блокування бази даних і проблем із втратою даних «розумних» міських послуг та застосунків. Видобування інформації передбачає вилучення фактів і структури логічної кореляції з вихідного набору даних, на відміну від самої інформації. Оптимізація запитів означає вибір оптимальної стратегії виконання запитів з мінімальними витратами та споживанням ресурсів системи. Алгоритми інтелектуального аналізу даних «розумного міста» виконують поглиблені та обширні запити до бази даних, щоб витягти шаблони та знання з комплексних

даних [4] зібраних в міському середовищі. Альтернативні методології, такі як XML та об'єктні бази даних, ніколи не досягали такого рівня популярності, як технологія RDBMS. Впродовж останнього десятиліття наука та онлайн-постачальники хмарних послуг поставили під сумнів універсальність технології магазину даних. Цей напрямок думок призвів до розробки нової альтернативної системи баз даних, відомої як NoSQL, що означає «Не тільки SQL». NoSQL описує використання веб-розробниками нереляційних баз даних [5]. У 1998 році вперше був використаний термін NoSQL [6], і конференція з нереляційних баз даних у Сан-Франциско привернула до нього більшу увагу. Рисунок 1.1 описує ключові аспекти баз даних NoSQL.

Рисунок 1.1 – Основні функції NoSQL

Ерік Брюер представив теорію CAP – узгодженості, доступності та терпимості до розподілу [7]. Основні характеристики теорії CAP подані в таблиці 1.1.

Таблиця 1.1 – Теорія CAP

Послідовність	**Доступність**	**Толерантність до розділу**
1	**2**	**3**
Узгодженість означає, що дані в базі даних залишаються узгодженими після виконання операції.	Доступність означає, що система не матиме простоїв (100% гарантія безвідмовної роботи).	Толерантність до розділів означає, що система продовжує функціонувати, навіть якщо зв'язок між серверами є ненадійним.

Продовження таблиці 1.1

1	2	3
Наприклад, після операції оновлення усі клієнти бачать однакові дані.	Кожен вузол (якщо не вийшов з ладу) завжди виконує запит.	Сервери можуть бути розділені на кілька груп, які не можуть спілкуватися один з одним.

Теоретично виконати всі три вимоги неможливо. Таким чином, CAP забезпечує суттєву потребу розподіленої системи, щоб слідувати двом із трьох елементів. Отже, усі поточні бази даних NoSQL підтримують різні комбінації C, A та P теорії CAP, як описано на рисунку 1.2.

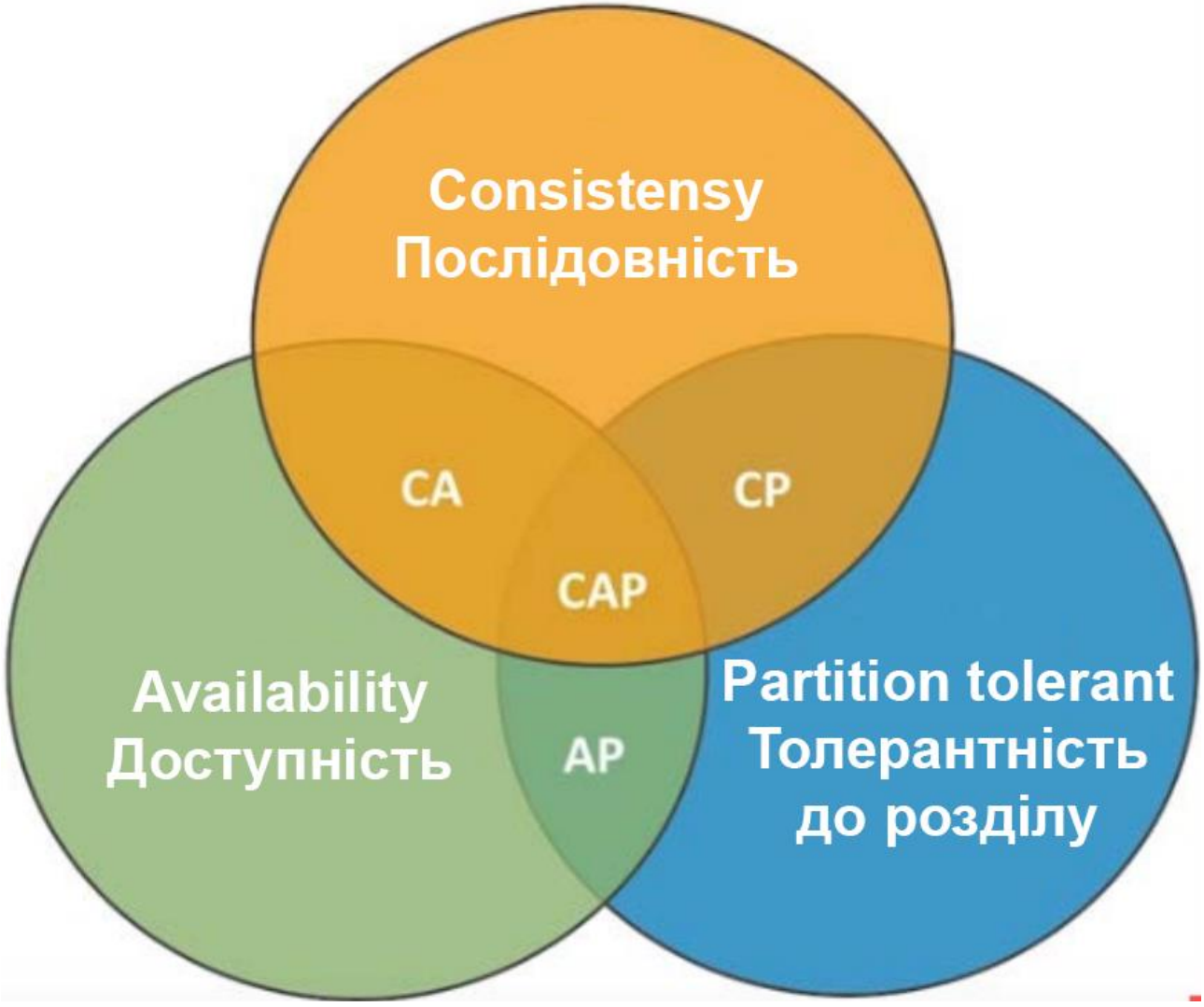

Рисунок 1.2 – Комбінації CAP

Розглянемо переваги баз даних NoSQL для «розумних міст»:

– Обсяг: дані в стані спокою – від терабайтів до ексабайтів існуючих даних для обробки.

– Швидкість: дані в русі – потокові дані, від мілісекунд до секунд для відповіді.

– Змінність: дані в багатьох формах – структуровані, неструктуровані, текстові тощо.

– Правдивість: сумнівні дані – невизначеність через затримку, обман, неоднозначність тощо.

– Не побудовано на основі таблиць і не використовує SQL для обробки даних.

– Схема містить ключ-значення, документ, стовпець, графік тощо.

– Альтернатива традиційним реляційним базам даних.

– База даних для обробки неструктурованих, безладних і непередбачуваних даних.

– Корисно для роботи з великими наборами розподілених даних.

Бази даних NoSQL відрізняються від реляційних баз даних і мають власні моделі та архітектури. Зберігаючи бази даних NoSQL у хмарі, необхідно дотримуватися обережності через різноманітність цих баз даних і необхідність взаємодії, переносимості та заходів безпеки. Постачальники хмарних послуг (CSP) [8] пропонують масштабованість, доступність і конфіденційність, але важливо зашифрувати конфіденційні дані користувачів «розумних міст» перед наданням доступу до CSP. Це може бути складно через різноманітність баз даних NoSQL. База даних як послуга (DBaaS) [9] – це хмарна платформа, яка перетворює традиційні архітектури на хмарні.

1.2 Стан проблемної області використання SQL та NoSQL баз даних для «розумних міст»

Поширення великих даних у «розумних містах» призвело до потреби в масштабованих системах, які можуть ефективно обробляти величезні обсяги

даних [10]. Реляційні бази даних, які базуються на SQL, можуть певною мірою керувати структурованими, напівструктурованими та неструктурованими даними, але мають обмеження щодо масштабованості. З іншого боку, бази даних NoSQL, які мають властивість BASE і можуть розширювати свою ємність зберігання по горизонталі, краще підходять для керування великими обсягами даних «розумних міст» і адаптації до змін у типі та структурі даних.

Існує чотири типи баз даних NoSQL – бази даних типу «ключ-значення», бази даних документів, стовпців і графіків – кожна зі своїми відмінними функціями та застосуваннями. Наприклад, графічна база даних NoSQL зберігає інформацію у вузлах, а не в таблицях, і зберігає асоціації (з'єднання) між вузлами, що вимагає постійного часу виконання. Це дає йому перевагу перед базами даних SQL при обробці сильно взаємопов'язаних даних.

MongoDB є прикладом бази даних NoSQL, яка ефективно керує величезними даними завдяки чудовим характеристикам масштабованості. Однак перетворення баз даних «розумних міст» OLTP на MongoDB може спричинити такі проблеми, як унікальні індекси, складені ключі, невідповідність даних і дублювання даних через складність їхньої схеми.

Іншою важливою проблемою при передачі хмарних сховищ «розумних міст» є захист особистої інформації користувачів. Поточні проекти постачальників хмарних послуг (CSP) «розумних міст» розробляються без урахування проблем сумісності та переносимості, що ускладнює пропонування та створення уніфікованого хмарного рішення для моделей NoSQL.

1.3 Критерії пошуку наукових публікацій щодо використання SQL та NoSQL баз даних у«розумних містах»

Критерії пошуку для первинних досліджень SQL та NoSQL баз даних для потреб «розумних міст» передбачають виявлення та збір літератури, яка відповідає критеріям включення та виключення. Щоб досягти цього, використовувалися різні методи пошуку, такі як пошук в електронній базі даних, ручний пошук і пошук за методом снігової кульки, а також пошук у відповідних

журналах і матеріалах конференцій. Для систематизації огляду наукової літератури будемо дотримуватись протоколу, запропонованого [11], відповідно до семи фаз, досліджених у [12]. Рисунок 1.3 ілюструє вибір пов'язаних досліджень [13].

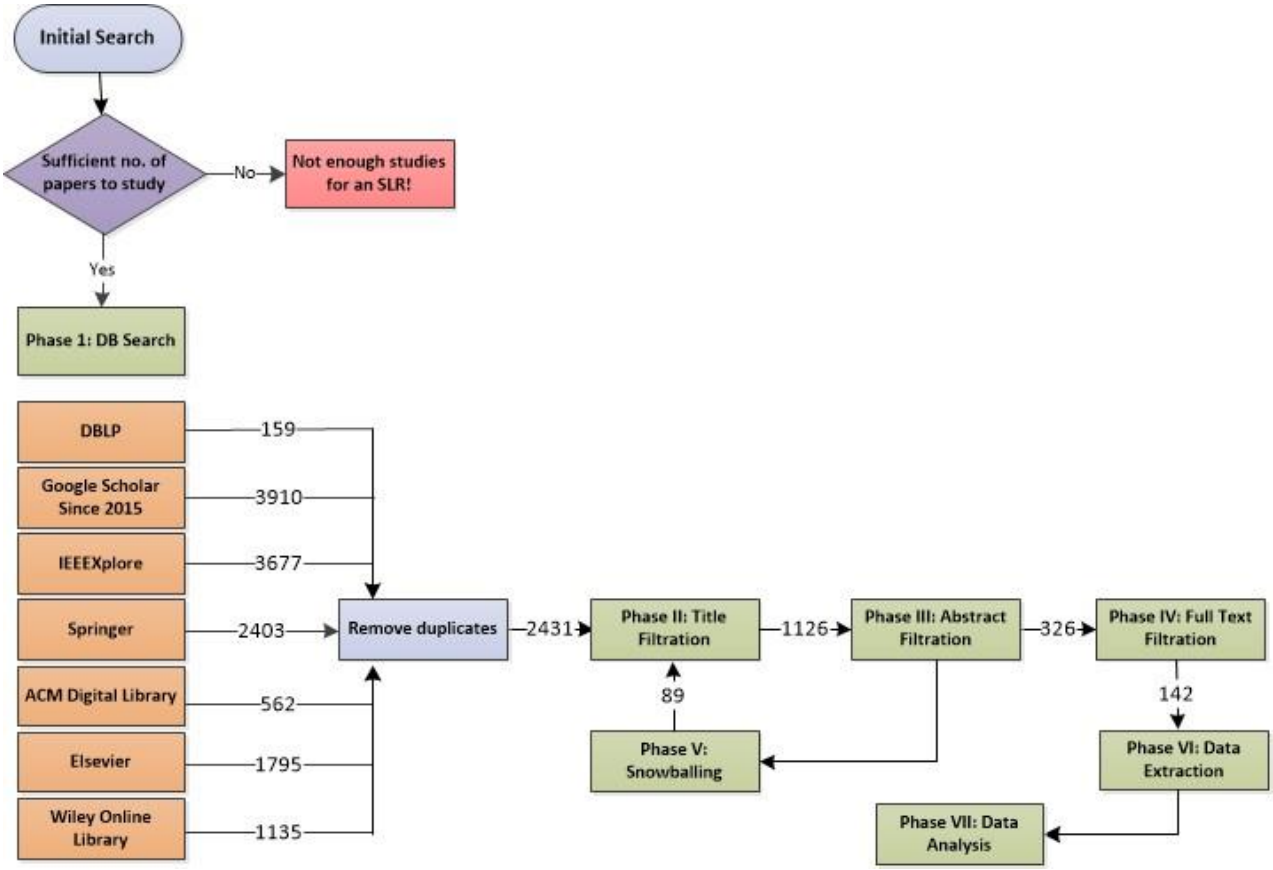

Рисунок 1.3 – Структура процесу наукометричного пошуку джерел [13]

Спочатку виконаємо пошук у базі даних, запропонований [14]. Рядки пошуку, згадані в розділі стратегії пошуку, використовувалися для категоризації та класифікації, тобто інструменти, методи та структура.

1.4 Задіяні пошукові ресурси

На першому етапі потрібно отримали набір асоційованих рядків пошуку. Для цього доцільно використати отриманий набір рядків, щоб знайти відповідні документи. Для пошуку пов'язаних статей були обрані найбільші наукометричні бази даних:

– DBLP.
– Цифрова бібліотека IEEExplore – ieeexplore.ieee.org.
– Google Scholar.
– Цифрова бібліотека ACM – dl.acm.org.
– Springer – Springerlink.com.
– Elsevier – sciencedirect.com.
– Інтернет-бібліотека Wiley – onlinelibrary.wiley.com.

Було знайдено понад тринадцять тисяч статей під час процесу пошуку за назвами статей, анотаціями та ключовими словами. Було відібрано понад дві тисячі чотириста статей, які відповідали критеріям включення та виключення після видалення дублікатів на першому етапі. Зворотна сніжна кулька на четвертій фазі призвела до відбору майже дав'яноста статей після неодноразової перевірки критеріїв включення та виключення на етапах від другого по четвертий.

1.5 Стратегія наукометричного пошуку матеріалів щодо використання SQL та NoSQL баз даних для «розумних міст»

Були створені різні комбінації пошукових запитів. Розроблені пошукові запити було запущено на згаданих пошукових ресурсах для ідентифікації пов'язаної літератури:

– «Smart City SQL і NoSQL».
– «Smart City SQL або NoSQL».
– «Розумне місто».
– «Розумне місто і системи керування базами даних».
– «Розумне місто і Реляційна база даних і база даних документів».
– «Розумне місто і Реляційна база даних або база даних документів».
– «Розумне місто і Реляційна база даних і база даних документів NoSQL».
– «Розумне місто і Реляційні бази даних і MongoDB».
– «Розумне місто і Реляційні бази даних або MongoDB».
– «Розумне місто і Порівняння Oracle і MongoDB».

– «Розумне місто і Порівняння баз даних SQL і NoSQL».
– «Розумне місто і Переваги MongoDB перед RDBMS».
– «Розумне місто і Реляційні та нереляційні бази даних».
– «Розумне місто і Переносність та сумісність хмарних даних».

1.6 Процес і критерії відбору наукових публікацій

На початковому етапі було включено всі пов’язані документи, застосовуючи критерії включення та виключення. На цьому етапі було оцінено якість і характеристики наукових статей відповідно до поставлених завдань та сформовано вибраний список. При цьому наукові публікації впорядковано за джерелами відповідно до:

– Загальна кількість документів.
– Назви робіт.
– Тези доповідей.
– Повне читання супутніх статей.

Потім було перевірено якість і вплив пов’язаних документів:

– Перевірено наукову статтю в каталозі, щоб уникнути повторення.
– Додано запис до остаточного каталогу документів

На наступному етапі було проведено ручний пошук за методом снігової кульки. Процес пошуку було ітеративно повторено декілька разів.

У таблиці 1.2 показано кількість пов’язаних документів, пов’язаних із зазначеними пошуковими запитами після кожної фази фільтрації.

Таблиця 1.2 показує, що спочатку було відібрано понад тисячу сто наукових статей на основі їхніх назв. Переглянувши їхні анотації, було обрано понад триста двадцять статей, а для повного читання було відібрано понад сто сорок статтей. З них майже дев’яносто було обрано методом сніжної кульки.

Таблиця 1.2 – Характеристики та критерії процесу наукометричного пошуку та відбору

Пошук \ Джерело	**На основі назви (другий етап)**	**На основі анотації (третій етап)**	**На основі повного читання статей (четвертий етап)**
Загальна кількість робіт	1126	326	142

Більшість наукових публікацій базувалися на емпіричних дослідженнях, які включають:

– Мету дослідження.

– Пов’язану літературу та підтримувані теорії.

– Гіпотетичне вимірювання.

– Запропонований метод, дизайн, підхід, розмір і збір даних.

– Аналіз результатів даних.

– Висновок.

1.7 Критерії включення наукових публікацій

Згідно з поставленими дослідницькими завданнями були включені наступні типи робіт:

– «IC1» відповідні та оглядові документи.

– «IC2» запропоновані нові методи та підходи, що стосуються запропонованої в [13] системи наукометричного пошуку.

– «IC3» ефективні методи дослідження, представлені в запропонованому дослідженні.

Щоб підвищити надійність і ефективність було переглянуто і досліджено вплив і методологію включених документів.

1.8 Критерії виключення наукових публікацій

Дослідницькі роботи були виключені на основі таких критеріїв:

– «EC1» документи, які не стосуються згаданої області.

– «EC2» невідповідні документи.

– «EC3» деякі статті на основі назви та анотації.

– «EC4» нерецензовані матеріали та документи.

– «EC5» статті, написані не англійською мовою, і дубльовані статті.

Щоб зменшити загрозу для надійності наукометричного пошуку, було повторно перевірено виключені документи відповідно до контрольного списку критеріїв виключення.

1.9 Збір і вилучення даних щодо наукових публікацій

Після збору понад ста сорока пов’язаних досліджень їх було переглянуто та виділено цінні дані, які задовольняли питання дослідження [15], зокрема, з кожної вибраної статті було отримано такі дані:

– Назва статті.

– Анотація роботи.

– Джерело паперу (журнал або конференція).

– Рік видання.

– Класифікація паперу (тип, сфера застосування).

– Спорідненість із тематикою кваліфікаційної роботи.

– Запропоновані цілі та проблеми дослідження.

– Анотація та метод.

Було вибрано більшість емпіричних статей. Емпіричні дослідження складалися з категорій оцінок статей, оцінок обговорень, експериментів та оглядів існуючих методів використання SQL і NoSQL для потреб «розумних міст».

1.10 Висновок до першого розділу

В першому розділі кваліфікаційної роботи освітнього рівня «Магістр» висвітлено актуальність дослідження предметної області використання SQL та NoSQL баз даних у «розумних містах». Описано стан проблемної області використання SQL та NoSQL баз даних для «розумних міст». Подано критерії пошуку наукових публікацій щодо використання SQL та NoSQL баз даних у«розумних містах». Розглянуто задіяні пошукові ресурси. Сформовано стратегію наукометричного пошуку матеріалів щодо використання SQL та NoSQL баз даних для «розумних міст». Описано процес і критерії відбору наукових публікацій. Подано критерії включення наукових публікацій. Наведено критерії виключення наукових публікацій. Розглянуто збір і вилучення даних щодо наукових публікацій.

2 АНАЛІЗ РЕЗУЛЬТАТІВ НАУКОМЕТРИЧНОГО ПОШУКУ ЩОДО ВИКОРИСТАННЯ SQL ТА NOSQL БАЗ ДАНИХ У «РОЗУМНИХ МІСТАХ»

2.1 Аналіз і класифікація результатів наукометричного пошуку

Інформація про кожну із знайдених наукових публікацій була впорядкована та зведена в табличній формі відповідно до питань:

1. Розгляд великих даних структурованих і неструктурованих даних «розумних міст».

2. Для чого потрібні NoSQL бази даних?

3. Чому база даних NoSQL слідує властивості BASE замість властивості ACID бази даних SQL?

4. Чи ефективно DBaaS забезпечує взаємодію та переносимість даних у різних базах даних NoSQL?

Під час аналізу було згруповано всі знайдені наукові публікації за категоріями та узагальнено висновки на три групи:

– методи дослідження;

– фази процесу дослідження;

– оцінка.

Згідно [16] слід послідовно оцінювати загрози процесу наукометричного пошуку. Ці загрози були розділені на різні групи:

– описова валідність;

– теоретична валідність;

– інтерпретаційна та узагальнена валідність;

– повторюваність [17].

2.2 Результати пошуку наукових публікацій

Підсумуємо наукову вибірку за роками публікацій, напрямком статті та кількістю вибраних досліджень із певної цифрової бібліотеки На основі

процедури відбору та критеріїв наукометричного пошуку були обрані дослідницькі статті. Згідно критеріїв пошуку було обрано обидва типи баз даних для рекомендованих методологій і досліджень. Окрім фактичних досліджень, також було виявлено опитувальні статті щодо використання баз даних SQL і NoSQL у «розумних містах». Після процедури відбору та оцінювання вибрані дослідження та публікації було розділено на три основні категорії. На рисунку 2.1 подано кругову діаграму категорій досліджень.

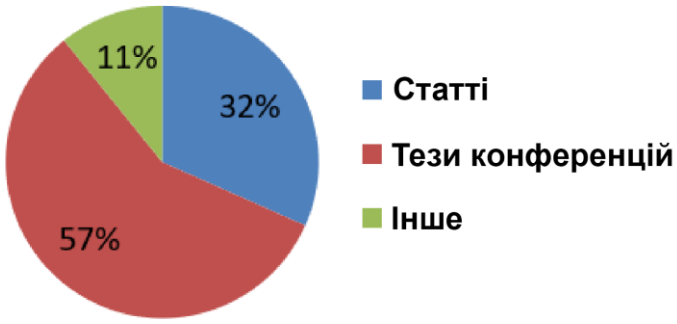

Рисунок 2.1 – Категоризація вибраних наукових публікацій [13]

Роки публікації вибраних досліджень, пов’язаних із базами даних SQL і NoSQL для потреб «розумних міст», коливаються між 2000 і 2022 роками. На рисунку 2.2 зображено кількість статей, що публікуються щорічно, при цьому слід зауважити, що бази даних NoSQL в «розумних містах» отримали більший розвиток після 2008 року.

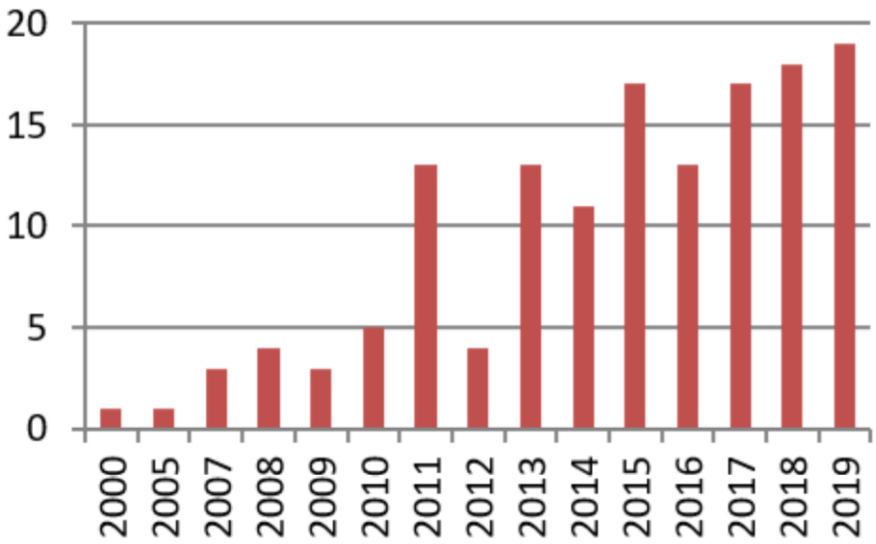

Рисунок 2.2 – Розподіл вибраних наукових публікацій по роках [13]

На рисунку 2.3 зображено кількість документів, підключених до реляційних баз даних і баз даних NoSQL «розумних міст» для кожної цифрової бібліотеки. Найбільш відповідні статті знаходяться в цифрових бібліотеках IEEE та Springer.

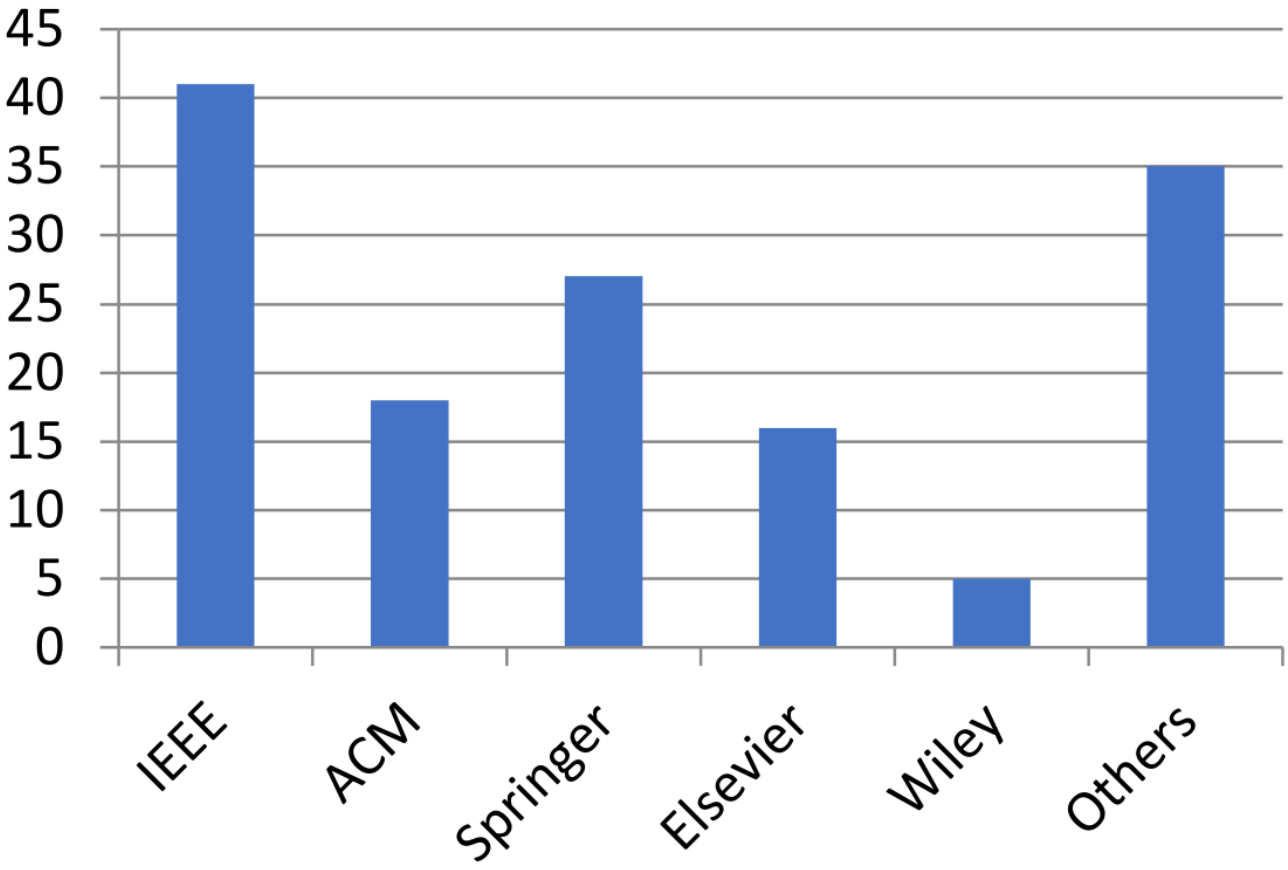

Рисунок 2.3 – Кількість документів на цифрову бібліотеку [13]

Інші можливості на рис. 2.3 включають статті, офіційні документи, розділи книг і технічні звіти від різних видавництв, включаючи Oracle, MIT, Academic Journal, SciTePress, IJACSA та IOPScience.

Обширне коло дослідників порівнювали продуктивність і властивості баз даних, наприклад MongoDB, MySQL, Cassandra, Couchbase, Oracle, SQL Server, PostgreSQL, Neo4j і Hbase впродовж усього вивчення вибраних досліджень.

На рисунку 2.4 зображено категорію паперу (журнал/матеріали конференції) конкретної цифрової бібліотеки з рисунка 2.3.

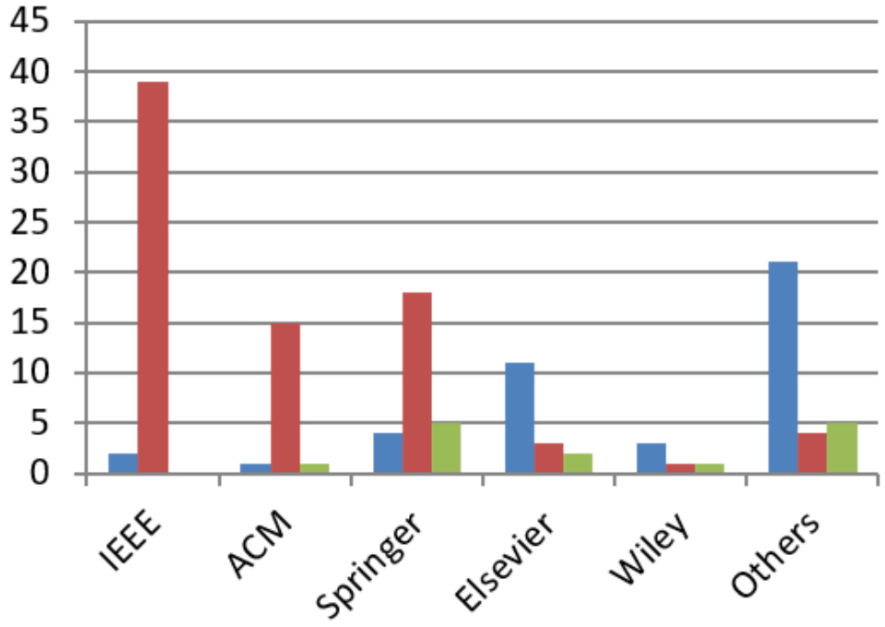

Рисунок 2.4 – Кількість категорій статей на джерело [13]

На рисунку 2.5 показано різноманітність баз даних, які використовуються для відбору загальної кількості дослідницьких робіт.

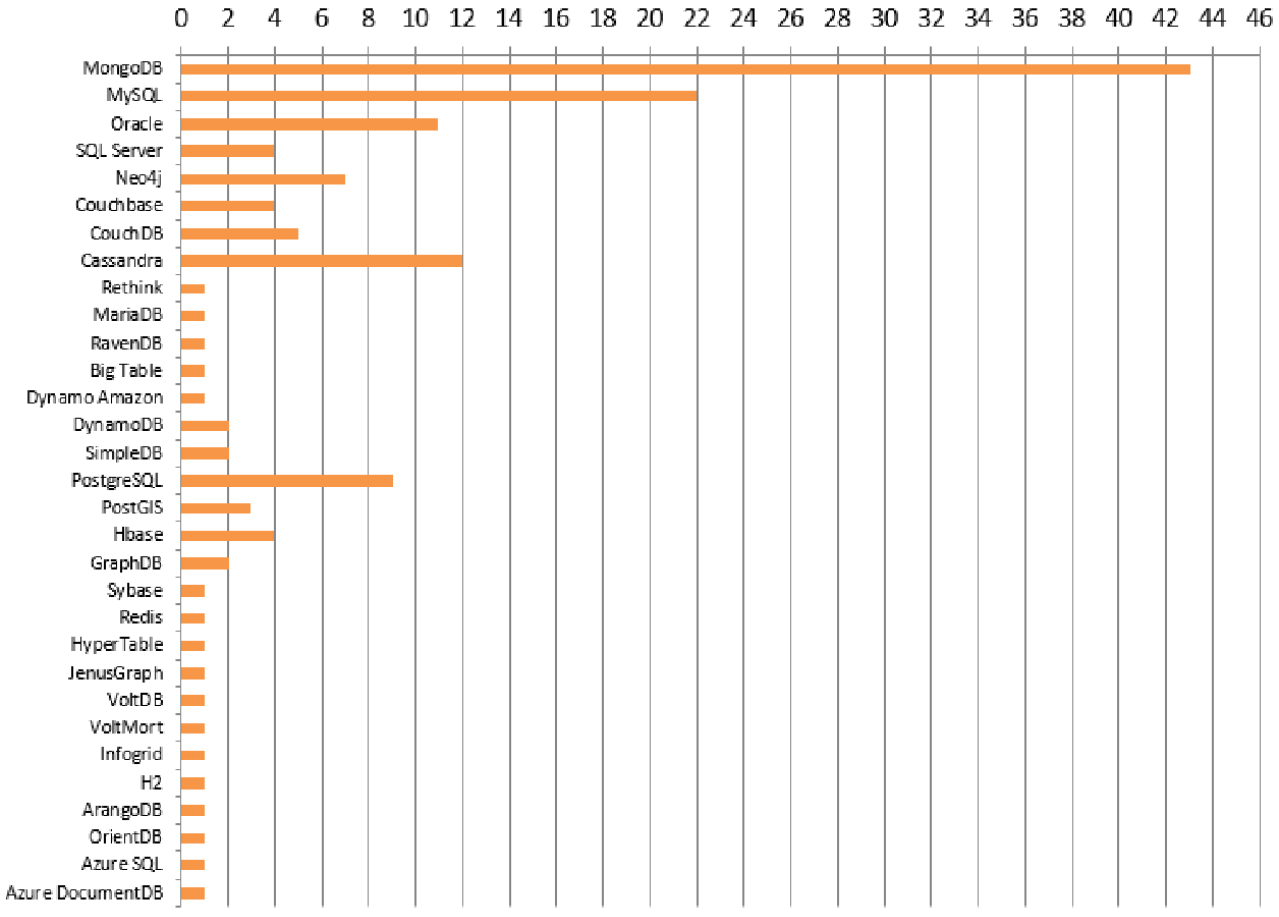

Рисунок 2.5 – Бази даних, використані в окремих дослідженнях [13]

Як видно на рисунку 2.5, більшість досліджень, які порівнювали та аналізували продуктивність і атрибути, для «розумних міст» використовували одну або кілька з таких баз даних: MongoDB, MySQL, Oracle, Cassandra, PostgreSQL, Neo4j, SQL server або CouchDB.

2.3 Аналіз результатів наукометричного пошуку

Проаналізовані опубліковані результати досліджень щодо використання SQL та NoSQL баз даних у «розумних містах» належать до категорій:

– порівнянь;

– оцінок;

– експериментів;

– категоризацій;

– діалогів;

– опитувань.

Статті, які, мали відношення до використання SQL та NoSQL баз даних у «розумних містах», були обрані з переліків наукової літератури та оцінені відповідно до критеріїв відбору.

Розгляд великих структурованих і неструктурованих даних:

– Для чого потрібні бази даних NoSQL?

– Чому бази даних NoSQL використовують властивість BASE замість властивості ACID бази даних SQL?

Моделювання бази даних допомогло передбачити типи даних, які в ній зберігатимуться, і спосіб їх зберігання.

NoSQL, абревіатура від «Not only SQL» [18], – це підхід до керування базами даних, який ефективний для обробки величезних обсягів неструктурованих даних і аналітики великих даних в проектах класу «розумне місто» [19]. З цими базами даних можна використовувати всі види різних мов запитів, і вони не дотримуються суворої, заздалегідь визначеної структури схеми. Однак впродовж декількох останніх десятиліть реляційні бази даних використовували галузевий стандарт мови SQL. Документно-орієнтовані бази

даних є підмножиною баз даних NoSQL. Бази даних, які зосереджені на зберіганні та отриманні документів, це MongoDB і CouchDB. Бази даних цього типу використовуються для зберігання та адміністрування даних в проектах класу «розумне місто», які переважно базуються на документах.

Складні формати даних JSON, BSON, XML і PDF використовуються для зберігання інформації в документоорієнтованих базах даних «розумних міст». І MongoDB, і CouchDB є безкоштовними та з відкритим кодом. Однак MongoDB [20] більше підходить для розподіленого налаштування та JSON. Автори [21] досліджували декілька різних функцій бази даних NoSQL. Однією з популярних баз даних, яка була створена спеціально з урахуванням JSON, є MongoDB [18], яка використовує мову програмування C++. MongoDB використовує динамічну схему [12] структури [1] замість попередньо визначених статичних документів.

Аналіз і пошук даних є швидкими та точними завдяки вдосконаленій обробці запитів, підтримці індексування та агрегації в пам'яті. Окрім забезпечення повної безпеки, він надає утиліти відновлення та резервного копіювання. Бази даних SQL і NoSQL «розумних міст» показані на рисунку 2.6 разом із відповідною структурою зберігання даних.

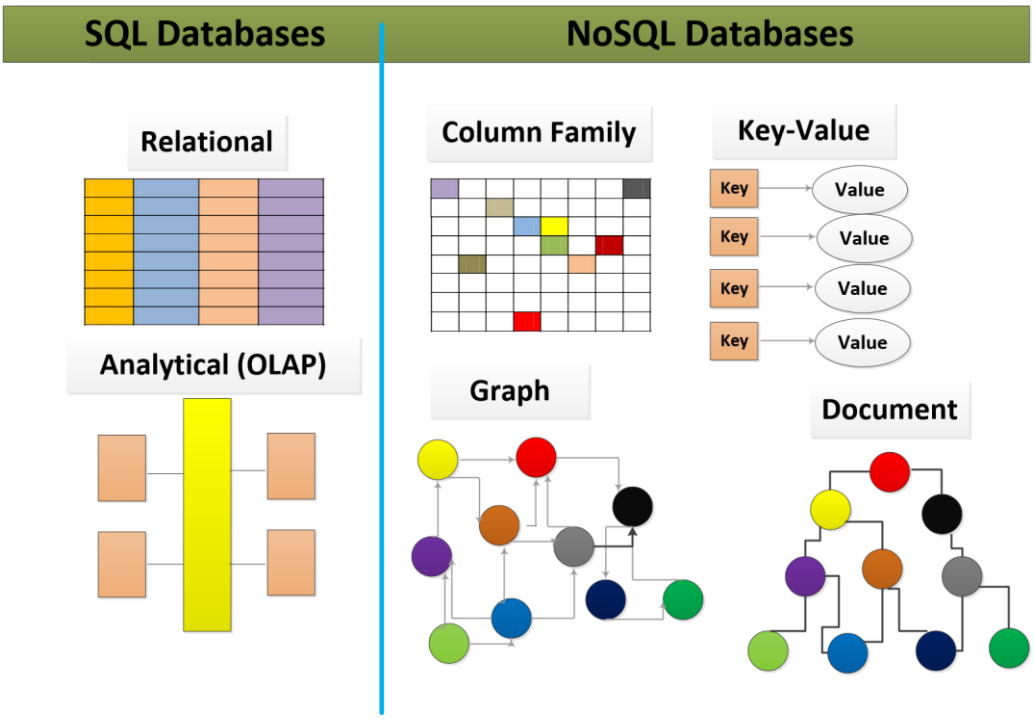

Рисунок 2.6 – Структури зберігання баз даних SQL і NoSQL «розумних міст» [13]

Бази даних SQL – Oracle, MySQL і SQL-Server та NoSQL – MongoDB, Neo4j, були предметом численних досліджень в контексті їх використання для потреб «розумних міст», що порівнювали їх структуру, дизайн і продуктивність [22]. Ці дослідження використовували запропоновані методи для ситуацій, пов'язаних із базами даних SQL і NoSQL в проектах класу «розумне місто», і аналізували результати. Бази даних NoSQL мають різні функції та програми в проектах класу «розумне місто». Бази даних NoSQL не можуть гарантувати властивість ACID через здатність масштабуватись горизонтально. У цьому плані база даних графів Neo4j [21] є відмінним варіантом.

Доведено, що бази даних графів [23] ефективно організовують і зберігають дані зі складними залежностями в проектах класу «розумне місто». Основним фокусом MongoDB є документи. MongoDB використовує формат BSON, який є відгалуженням JSON.

Маючи величезні дані в проектах класу «розумне місто», MongoDB працює добре і неодноразово демонстрував себе. Дослідження [23] порівнювало базу даних NoSQL MongoDB з реляційною базою даних Postgre SQL для соціальної мережі та служби даних «розумних» давачів потоку. У сценаріях, де порівнювали дві бази даних «розумних міст», MongoDB зайняла перше місце [24].

На відміну від систем керування реляційними базами даних (RDBMS), база даних NoSQL [25] добре підходить для використання в кластері «розумних міст», де її гнучка та потужна архітектура може вмістити величезні обсяги даних у різноманітних формах. MongoDB і MySQL порівнювалися з канонічною системою управління базами даних у [26]. MongoDB перевершив MySQL у пошуку та вставці даних в проектах класу «розумне місто».

Грунтуючись на проведених порівняннях, було зрозуміло, що NoSQL MongoDB є найкращим варіантом, ніж MySQL, при обробці великих обсягів даних «розумних міст» [27]. Для статей було проведено поглиблене дослідження передумов для ефективного управління величезними обсягами неструктурованих даних «розумних міст» [28]. Вони віддали перевагу використанню бази даних NoSQL MongoDB замість використання MySQL

RDBMS. Реляційна та нереляційна моделі баз даних були детально розглянуті авторами [29].

Залежно від аналізу база даних NoSQL MongoDB перевершила базу даних MySQL в проектах класу «розумне місто». Вони припустили, що для невеликого набору даних із базовими запитами MySQL є ефективним, тоді як для великих наборів даних «розумних міст» із складними запитами більш прийнятним рішенням є MongoDB. Дослідними також порівнювали продуктивність Oracle RDBMS з базою даних NoSQL MongoDB. У результаті свого дослідження вони дійшли висновку, що бази даних NoSQL не є життєздатною альтернативою базам даних SQL. Вони [30] стверджували, що компанії можуть вибрати базу даних, яка відповідає їхнім потребам, враховуючи специфіку своєї компанії.

Що стосується масштабованості даних, високої продуктивності та доступності даних в проектах класу «розумне місто», було знайдено обширний перелік публікацій [31], в яких обговорюється перехід від реляційної бази даних до бази даних NoSQL «розумних міст», а також інші статті [31] на цю тему.

Завдяки масштабованості, цілісності, розповсюдженню, безпеці [32] і настроюваному дизайну MongoDB добре підходить для обробки величезних обсягів даних «розумних міст». Згідно з цитованими роботами, перехід від реляційної бази даних до бази даних NoSQL MongoDB в проектах класу «розумне місто» – це шлях. Для зберігання інформації MongoDB використовує формат двійкової нотації об’єктів JSON (BSON). Оскільки для неї не потрібні з’єднання, як для реляційних баз даних, MongoDB здатна ефективно зберігати та отримувати велику кількість документів в одній колекції «розумних міст».

MongoDB досить універсальний, щоб обробляти та обробляти дані в широкому спектрі форматів, включаючи структуровані, напівструктуровані та неструктуровані дані «розумних міст» [33]. Реляційні бази даних були створені з явною метою обробки впорядкованих даних, і тому вони можуть обробляти певну кількість даних, які є вертикальними за своєю природою. Щоб було зрозуміло, бази даних NoSQL не призначені для заміни реляційних баз даних в проектах класу «розумне місто», але їх підвищена продуктивність і корисність означають, що вони мають місце поряд з реляційними базами даних у деяких

контекстах. Згідно з дослідженням [34], система реляційної бази даних погано масштабується з величезними обсягами даних «розумних міст». Автоматичне відображення з MySQL RDBMS в базу даних MongoDB NoSQL, використовуючи метадані «розумних міст», збережені в MySQL RDBMS, було запропоновано в статті [35]. Бази даних NoSQL не обіцяють запропонувати ту саму атомарність, послідовність, ізоляцію та довговічність, які є ознаками систем управління реляційними базами даних [36].

Функція BASE [37] спостерігається в базах даних NoSQL. NoSQL [38] – це загальний термін для набору технологій, які мають спільні цілі щодо доступності даних, ефективного масштабування даних, ефективного керування сховищем і покращеної продуктивності «розумних міст». Дослідження [21] розглядає мотиви для переходу від систем управління реляційними базами даних до нереляційних баз даних, орієнтованих на документи. Тим часом аспекти ACE – доступність, узгодженість та ефективність системи великих даних «розумних міст» досліджувалися в обох базах даних [39].

Дослідження теми автоматичного перетворення схеми з SQL на NoSQL «розумних міст» подано в [40].

У статті [41] порівнювалася ефективність кількох баз даних NoSQL, включно з Couchbase, MongoDB, Re-thinkDB і in-memory, при використанні з даними «розумних міст» реального світу. Плануючи запропоновані випробування, автори враховували кілька факторів, наприклад час відгуку різних баз даних «розумних міст». У дослідженні [42] було проведено порівняльний аналіз продуктивності Oracle NoSQL і MongoDB. Бази даних вивчалися з різних точок зору, включаючи їх модель зберігання, масштабованість, паралельність і реплікацію. Автори дійшли висновку, що в цілому MongoDB був більш популярним в проектах класу «розумне місто», ніж пропозиція Oracle NoSQL.

Згідно з рейтингом DBEngine.com, MongoDB є п’ятою найкращою базою даних NoSQL, тоді як Oracle NoSQL займає сімдесят восьме місце. У статті [43] стверджується, що порівняно з MapReduce MongoDB агрегація RDBMS Oracle працює краще. З іншого боку, щодо швидкості відповідей на запити MongoDB перевершив Oracle RDBMS.

Відповідно до [28], агрегація Oracle була кращою за агрегацію MongoDB для робочих навантажень SUM, COUNT і AVG. У роботі [30] використовувався набір даних із приблизно 500 000 записами, чого було недостатньо для аналізу великих даних. Схема процесу оператора SQL Select в Oracle 11g RDBMS зображена на рисунку 2.7.

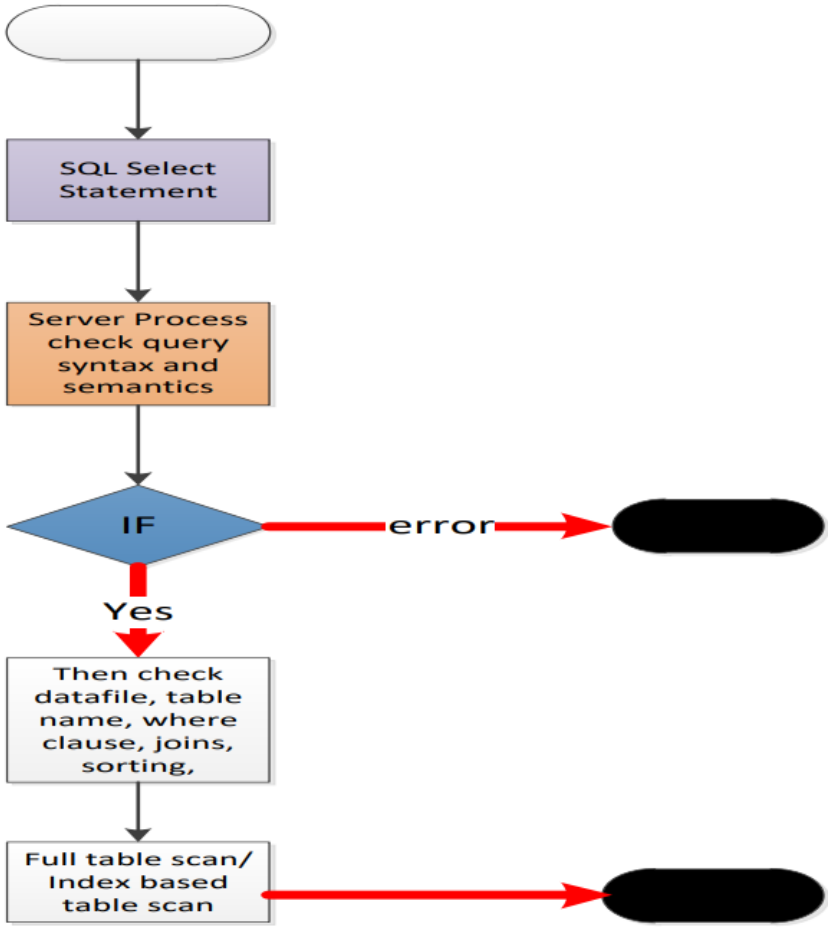

Рисунок 2.7 – Блок-схема процесу оператора SQL Select [13]

Отримання даних MongoDB в проектах класу «розумне місто» є простим і легким завдяки структурі піддокументів і не вимагає перевірки обмежень або будь-яких пунктів, на відміну від оператора SQL Select Oracle 11 g RDBMS. Потік процесу оператора SQL Insert показано на рисунку 2.8.

Порівняно з RDBMS Oracle, вставка MongoDB є швидшою, оскільки не потрібно перевіряти кроки, показані на рисунку 2.8 оператора SQL Insert.

Навпаки, MapReduce – це модуль програмування [44], яка добре працює в сценаріях розповсюдження та є більш ідеальним для аналізу великих даних [45] порівняно з базовим агрегуванням [43] у кластерних багатосерверних

контекстах. Перед початком етапу карти операції MapReduce в проектах класу «розумне місто» можуть виконувати будь-яке довільне сортування та обмежують документи в одній колекції, яку вони використовують як вхідні дані.

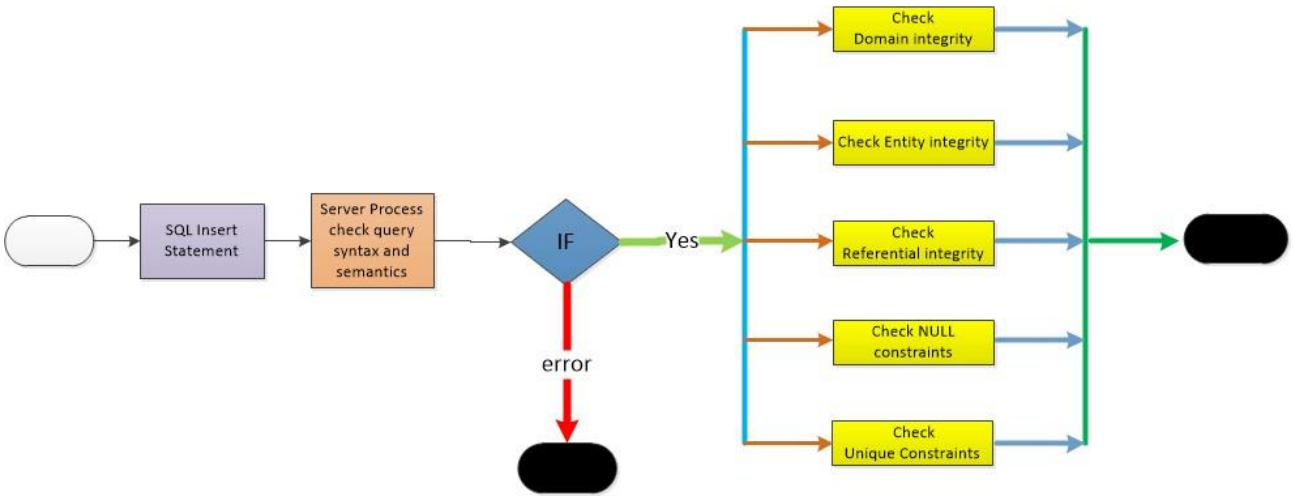

Рисунок 2.8 – Блок-схема процесу оператора SQL Insert [13]

Для роботи алгоритму MapReduce в проектах класу «розумне місто» потрібні дві фази MapReduce, «Map» і «Reduce». MongoDB використовує фазу карти для кожного документа, який подається в MapReduce. Функція map повертає пару критичних значень. MongoDB використовує процес під назвою «зменшення», щоб зібрати та стиснути агреговані дані, а потім MongoDB зберігає результати в колекції «розумного міста». Наприклад, у наборі даних про злочини в Чикаго функція карти обчислює всі злочини проти кожного дня, а потім функція зменшення бере день як ключ і витягує відповідні пари «Ключ значення». Фреймворк MapReduce–Merge був розроблений авторами [46] шляхом включення операції злиття в архітектуру MapReduce. Продуктивність MapReduce в проектах класу «розумне місто» була збільшена та змогла обчислити реляційну алгебру завдяки операції злиття та змогла обробляти дані в кластері. Щоб максимізувати продуктивність MapReduce і підвищити продуктивність запитів кластера, інші запропонували структуру MRShare [47]. Таким чином, агрегація Oracle RDBMS перевершила MongoDB в агрегації MapReduce.

За останнє десятиліття Oracle RDBMS широко використовувалася компаніями будь-якого розміру. Через архітектуру абстрактної системи з одним зображенням СУБД SQL погано справлялися з обробкою неструктурованих даних.

Дослідження [48] вивчало декілька особливостей і класифікацій баз даних NoSQL, що працюють поверх Hadoop. Воно продемонструвало, що бази даних NoSQL можна розбити на підкатегорії «розумних міст» відповідно до таких факторів, як масштабованість і тип даних, що підключений, або документ. В таблиці Таблиця 3 порівнює Oracle RDBMS з MongoDB і перераховує їхні основні функції.

Таблиця 2.1 – Характеристики MongoDB і Oracle RDBMS.

Колонка	**Поле**
Щоб отримати повну інформацію про одну особу, громадянина або клієнта, потрібні кілька об’єднань, що спричиняє повільний доступ до даних.	Доступ до даних дуже швидкий, оскільки всі дані зберігаються в одному документі
Вертикальна масштабованість	Горизонтальна масштабованість
Складний шардинг даних	Шардинг даних стає простішим
Розглядає ефективність зберігання даних	Враховує швидкість, продуктивність, час розробника
Добре працює в агрегації	Краще працює під час пошуку
Читання/запис не виконується дуже швидко в аналітиці великих даних	Виконує читання/запис дуже швидко завдяки функції відображення пам’яті в аналітиці великих даних.
Підтримує реляційну алгебру [38]	Підтримує реляційну алгебру

Згідно з авторами [49], в проектах класу «розумне місто» спостерігається помітне збільшення кількості геопросторових і геолокованих даних. Було розроблено багато програм для обробки та використання геопросторових даних,

і це стосується багатьох різних галузей міського життя, управління надзвичайними ситуаціями, археологія, Інтернет речей та «розумні» послуги. Для ефективної обробки великих обсягів географічних даних «розумних міст» [50] потрібна потужна система керування базами даних.

При роботі з величезними обсягами даних бази даних NoSQL, а не бази даних SQL, є найкращим варіантом для веб-застосунків «розумних» міських послуг [51]. Геопросторовими даними та даними про місцезнаходження в проектах класу «розумне місто» також керували за допомогою баз даних NoSQL [52]. У багатьох звітах [53] щодо «розумних міст» було запропоновано, що бази даних NoSQL здатні ефективно обробляти величезні обсяги неструктурованих міських даних. Дослідження [54] повідомляє, що дослідники вперше дізналися про просторові дані в основоположних текстах Географічних інформаційних систем. У дослідженні було досліджено дві проблеми з ефективною обробкою даних геопросторових запитів [55]. По-перше, звичайні методи оптимізації були недостатніми для вирішення географічних запитів. Взявши до уваги велику аналітику в проектах класу «розумне місто», стало зрозуміло, що всі методи та підходи до географічних запитів, випробувані досі, були недостатніми для величезних обсягів даних. На відміну від традиційних текстових даних «розумних міст», які містять лише обмежений набір якостей, геопросторові дані пропонують широкий спектр додаткових функцій. Широка обробка геопросторових даних, згідно з кількома дослідженнями [56], потребує добре налагоджених методів обробки величезної кількості наявних геопросторових міських даних. У статті [57] показано, що добре відомі системи RDBMS стикаються з численними труднощами при спробі аналізу географічних даних «розумних міст». Використовуючи порівняльний аналіз продуктивності, автори [52] розглянули базу даних документів NoSQL MongoDB порівняно з системою керування реляційною базою даних PostGIS. Як показали їхні тести, MongoDB працює краще, ніж PostGIS при обробці геопросторових даних «розумних міст».

Модель просторового зберігання даних Oracle [58] в проектах класу «розумне місто» мала два основних компоненти: розташування та форму. Моделі зберігання, такі як index Engine і Geometry Engine, використовувані для аналізу

запитів, використовували тип даних SDO_GEOMETRY. Для служб визначення місцезнаходження в проектах класу «розумне місто» використовувався геокодер для перетворення адреси в інформацію SDO_GEOMETRY. Для візуалізації використовувалися Oracle Maps і Map Viewer.

SQL Server 2016 та база даних SQL Azure і його хмарна версія включають різноманітні геофункції для аналітики геопросторових даних, як і просторову базу даних SQL Oracle, а бази даних NoSQL, наприклад, Azure DocumentDB і MongoDB, містять географічні можливості. Парадигма «База даних як послуга» (DBaaS), яка використовується базою даних Azure SQL, також має такі ж функції сервера Microsoft SQL і надає хмарні служби для потреб проектів класу «розумне місто». PostgreSQL – це безкоштовна відкрита система реляційних баз даних. Для порівняння, PostGIS є розширенням PostgreSQL, яке служить просторовою базою даних для роботи з геопросторовою інформацією. База даних NoSQL Azure DocumentDB створена компанією Microsoft і забезпечує ті самі географічні операції та функції в проектах класу «розумне місто», що й MongoDB. MongoDB підтримує стандартний формат даних GeoJSON. Автори [59] провели аналіз ефективності географічних даних. Згідно з їхніми дослідженнями, Azure DocumentDB в проектах класу «розумне місто» була швидшою, ніж база даних Azure SQL, але менш масштабованою, ніж база даних Azure SQL. Перелік підтримуваних геопросторових обєктів популярних баз даних SQL і NoSQL подано в таблиці 2.2.

Таблиця 2.2 – Геометричні об'єкти, що підтримуються базами даних SQL і NoSQL

Oracle	**PostGIS**	**Azure SQL**	**MongoDB**	**DocumentDB**
1	**2**	**3**	**4**	**5**
Point, LineString, Багатокутник, MultiPoint,	Point, LineString, Багатокутник, MultiPoint,	Point, LineString, Багатокутник, MultiPoint,	Point, LineString, Багатокутник, MultiPoint,	Point, LineString, Багатокутник, MultiPoint,

Продовження таблиці 2.2

1	2	3	4	5
MultiLinePoint, MultiPolygon, GeometryCollection	MultiLinePoint, MultiPolygon, GeometryCollection	MultiLinePoint, MultiPolygon, GeometryCollection	MultiLinePoint, MultiPolygon, GeometryCollection	MultiLinePoint, MultiPolygon, GeometryCollection

Перелік підтримуваних геометричних функцій популярних баз даних SQL і NoSQL подано в таблиці 2.3.

Таблиця 2.3 – Геометричні функції, що підтримуються базами даних SQL і NoSQL

Oracle	PostGIS	Azure SQL	MongoDB	DocumentDB
For geometry instances, Oracle has the support of Open Geospatial Consortium (OGC)	For geometry instances, PostGIS has the support of Open Geospatial Consortium (OGC)	For geometry instances, Azure SQL has the support of Open Geospatial Consortium (OGC)	Inclusion, Intersection, Distance/ Proximity	Inclusion, Distance/ Proximity

2.4 Моделювання даних NoSQL MongoDB для потреб «розумних міст»

На рисунку 2.9 зображено шард-вузли, сервери конфігурації та сервери маршрутизації, або mongo, які складають архітектуру MongoDB [60].

Дані проектів класу «розумне місто» зберігаються в сегменті, і кластер MongoDB неможливо побудувати без одного або кількох шардів. Коли вузол виходить з ладу, інформація для цього фрагмента зберігається його репліками. Транзакції даних, читання та запис, визначають, який шард буде використовуватися.

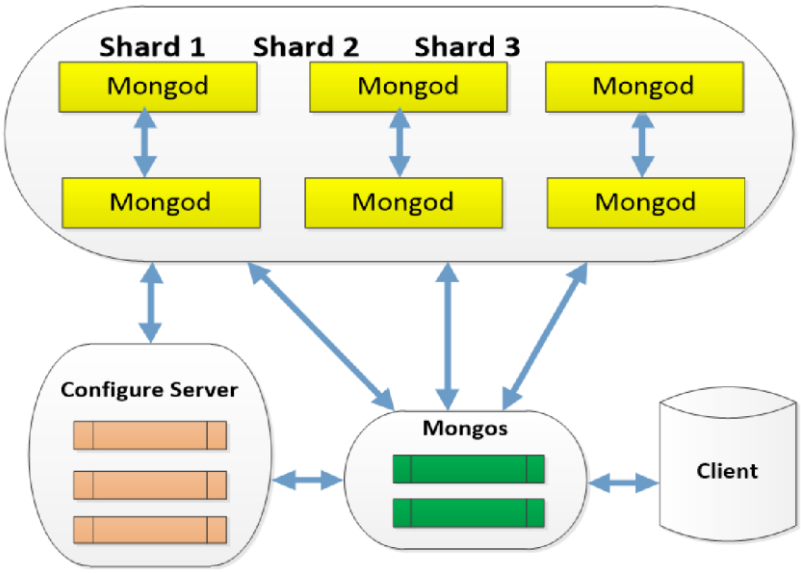

Рисунок 2.9 – Архітектура MongoDB [13]

Один або кілька серверів використовуються реплікованим вузлом, а вторинний вузол моделюється як вихідний сервер. Якщо основний сервер вийде з ладу, один із резервних серверів перейде на роботу. Сервер є центром, навколо якого відбуваються всі операції читання та запису. Зрештою, кластер буде синхронізований з усіма розподіленими транзакціями читання.

Набір серверів конфігурації в кластері «розумного міста» відповідає за зберігання метаданих. Шарди даних ідентифікуються цими серверами, які також передають, яка частина даних до якого шарду належить. Служба обслуговування клієнтів вимагає, щоб або сервери маршрутизації, або MongoDB виконали дію. Перш ніж отримати підтвердження від клієнта, MongoDB призначає кожне завдання користувача відповідному сегменту на основі типу завдання, а потім об’єднує отримані дані. Оскільки Mongos [19] не мають громадянства, їх можна використовувати в розподілених умовах проектів класу «розумне місто».

Файли з відображенням пам’яті використовуються MongoDB, щоб максимально використовувати доступну пам’ять, що, у свою чергу, підвищує продуктивність проектів класу «розумне місто». Індексація в базі даних MongoDB [61] використовує B-дерева. У кластері MongoDB [62] користувач може претендувати на право власності на певну колекцію розділів за допомогою шардового ключа.

2.5 Висновок до другого розділу

В другому розділі кваліфікаційної роботи проведено аналіз і класифікацію результатів наукометричного пошуку. Подано результати пошуку наукових публікацій. Проаналізовано результати наукометричного пошуку. Виконано моделювання даних NoSQL MongoDB для потреб «розумних міст».

3 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ВИКОРИСТАННЯ РІЗНИХ SQL І NOSQL БАЗАХ ДАНИХ «РОЗУМНИХ МІСТ»

3.1 Ефективність DBaaS, взаємодія та переносимість даних у різних NoSQL базах даних «розумних міст»

У попередніх розділах було проведено поглиблене дослідження літератури про архітектуру DBaaS для потреб проектів класу «розумне місто». Аналіз і вилучення даних показують, що хмарний підхід DBaaS в проектах класу «розумне місто», розроблений для реляційних баз даних, не є оптимальним для баз даних NoSQL. Час і зусилля експертів можна скоротити, а безпеку можна покращити, якщо вчені та дослідники зможуть знайти уніфіковане DBaaS [63] рішення для баз даних SQL і NoSQL. Завдяки стандартним API (API) також немає потреби перепроектовувати програми для використання різними проектами класу «розумне місто». Портативність даних і сумісність між різними хмарними провайдерами є основними перешкодами, які необхідно подолати. Інтероперабельність визначається по-різному кожною з трьох використовуваних в проектах класу «розумне місто» парадигм: IaaS, PaaS і SaaS [64]. Відкриті стандарти [65] допомагають пом’якшити проблему сумісності, однак рішення зосереджено саме на рівні IaaS. При передачі інформації між різними хмарними провайдерами для потреб проектів класу «розумне місто» зазвичай необхідні уніфіковані API [66]. Немає єдиної моделі зберігання даних, яка використовується всіма хмарними службами в проектах класу «розумне місто». Коли розробник переходить від одного CSP до іншого, дані передаються відповідно до архітектури високого рівня, зображеної на рисунку 3.1.

Бази даних [67], які є одночасно узгодженими та пов’язаними, а також їх ефективне управління [68] є ключовими та критичними проблемами «розумних міст» в епоху сучасних інформаційних технологій.

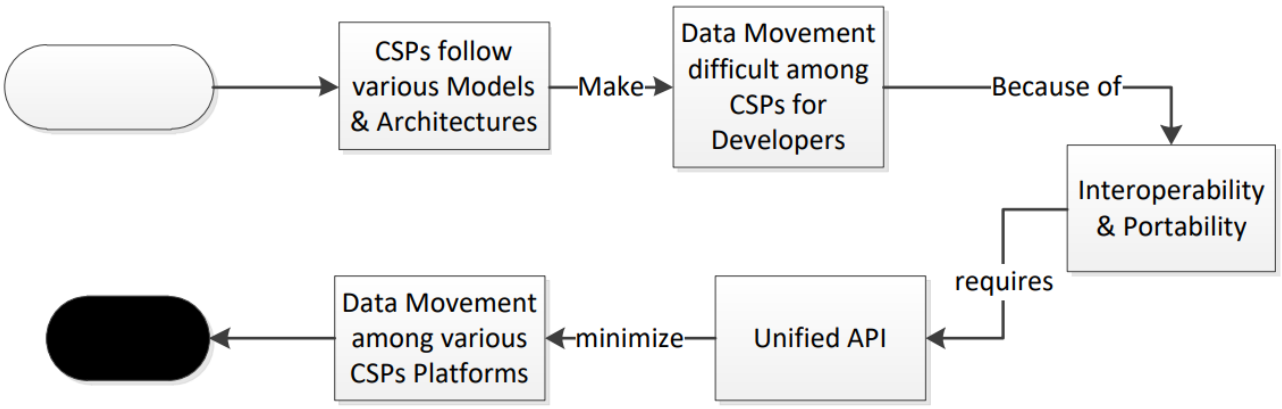

Рисунок 3.1 – Переміщення даних всередині CSP [13]

Основними особливостями системи баз даних в проектах класу «розумне місто» є гарантія постійного сховища даних щодо незалежності даних від базового фізичного носія, а також можливості обробки запитів, що забезпечуються декларативними запитами (DBS).

У секторі баз даних для потреб проектів класу «розумне місто» спостерігається велика різноманітність підходів до різних областей. Кілька DMS, включаючи ACID, OOM, XML і сховища даних, засновані на реляційній моделі даних. Однак атрибут BASE [37] підтримується в базах даних NoSQL, які в основному призначені для обробки великих обсягів даних.

Постачальники хмарних послуг пропонують клієнтам нові хмарні послуги [69] та можливості за низькою ціною та з високою ефективністю для потреб проектів класу «розумне місто». Однак декілька постачальників хмарних послуг пропонують однакові функції, використовуючи різні реалізації та інтерфейси користувача [70], що неминуче спричиняє проблеми з сумісністю [71], несумісністю та портативністю. Це складні проблеми для провайдерів хмарних послуг під час впровадження та полегшення хмарних технологій [72] для потреб проектів класу «розумне місто». Інтероперабельність IaaS, PaaS і SaaS в проектах класу «розумне місто» – це різні терміни з різним значенням і застосуванням у сфері хмарних сервісів [64]. Хмарні користувачі повинні бути переміщені з одного CSP до іншого з таких причин [65]:

– високі показники простою або відмови;
– розірвання контракту;
– зміни корпоративного плану;

– кращі альтернативи з низькою вартістю;

– юридичні труднощі.

Клієнти, які використовують кілька хмарних постачальників, не можуть легко перенести свої дані між ними. Моделі хмарних сервісів прагнуть контролювати можливості клієнта, оскільки їхні ключові архітектури не сумісні. Ця проблема в проектах класу «розумне місто» вирішується за допомогою блокування від постачальника, що створює значну проблему безпеки для хмарних моделей [65]. Портативність даних [73] покращиться, оскільки все більше постачальників хмарних послуг (CSP) приймають відкритий стандарт для вирішення проблем сумісності даних «розумних міст». У результаті розробнику може бути складно забезпечити узгодженість даних і програм у різних хмарних службах.

Відкритий формат віртуалізації (OVF), інтерфейс керування хмарною інфраструктурою (CIMI), інтерфейс відкритого хмарного обчислення (OCCI) та інтерфейс керування хмарними даними (CDMI) досліджувалися як потенційні рішення для керування сумісністю між CSP. Ці стандарти підтримують лише рівень IaaS. Подібним чином, багато ініціатив, таких як MOSAIC, MODACLOUDS і Clous4SOA [74], були створені для вирішення проблеми семантичної сумісності на рівні PaaS в проектах класу «розумне місто». Наявність кількох API для кожного PaaS в проектах класу «розумне місто» означає, що ініціативи не можуть самостійно вирішити проблему сумісності.

Як і очікувалося, стандарт CIMI сумісний з API IaaS і допомагає зменшити ступінь сумісності, необхідний користувачам хмари та їхнім постачальникам інфраструктурних послуг. Використовуючи OCCI, сумісність між CSP може бути зменшена, водночас одержуючи адекватні результати. Оскільки ці стандарти не включають функції сумісності в їх базові архітектури, вони не широко використовуються в проектах класу «розумне місто» різними CSP. Крім того, існує проблема з доступністю стандартизованих інтерфейсів та API.

Шаблони проектування, хмарні проміжні інфраструктури, хмарна платформа надання послуг (SDCP) в проектах класу «розумне місто», а інструменти міграції були розроблені іншими вченими [75] для полегшення

переміщення даних між різними хмарами. Незважаючи на економію часу користувачів, ці методи не вирішують проблему переносимості між різними CSP, яка виникає через необхідність вивчення та впровадження кількох API. Amazon Web Service (AWS), Microsoft Azure та Google App Engine (GAE) є прикладами постачальників хмарних послуг, які допомагають в проектах класу «розумне місто» створювати та запускати хмарні програми. Крім того, вони надають хмарну платформу Database as a Service (DBaaS) для підтримки розробників баз даних. Перенесення даних у DBaaS, яке може відбуватися між базами даних SQL і NoSQL [76], а також усередині кожного типу бази даних NoSQL, в проектах класу «розумне місто» потенційно може бути проблематичним через проблему сумісності. Багато типів баз даних NoSQL дотримуються несумісних форматів зберігання та парадигм даних. Існує потреба в стандартизованих API для потреб проектів класу «розумне місто», які можуть регулювати переміщення даних між різними платформами хмарного зберігання.

Для розміщення своїх даних і застосунків розробники програмного забезпечення можуть скористатися перевагами DBaaS [77], високомасштабованої та доступної серверної хмарної служби. Як наслідок, DBaaS зараз є найпривабливішим варіантом для хмарних клієнтів [78]. База даних як послуга (DBaaS) [79] – це хмарна служба, яку пропонують хмарні постачальники (CSP), яка полегшує перехід від традиційної архітектури бази даних до архітектури хмарної бази даних «розумного міста». SaaS полегшує віддалений доступ до комп’ютерних програм та їхніх функцій і можливостей для потреб проектів класу «розумне місто». PNUTS, HBase, SimpleDB і Google BigTable – це деякі інші хмарні служби баз даних. Потенційно структура DBaaS може добре працювати з традиційними базами даних. Однак продуктивність DBaaS знижується в таких сферах, як узгодженість даних, конфіденційність, цілісність, доступність і відсутність безпеки через різні підходи проектів класу «розумне місто», яких дотримуються численні бази даних. Якби конфіденційність і безпека даних були гарантовані в проектах класу «розумне місто», модель аутсорсингу, відома як DBaaS, може бути фінансово успішною. У дослідженні [80] було запропоновано консультанта з віртуалізації для

приватних хмар баз даних таблиць «розумного міста». Хмарні обчислення дають змогу інтегрувати численні програми в структуру самого сервісу. Масштабованість і адаптивність хмарних обчислень коштує нижче, ніж будь-коли раніше.

Щоб забезпечити більшу сумісність даних, портативність і безпеку під час перенесення даних, у статті [63] досліджено дві уніфіковані структури API, CDPort і Se-cloudDB, для баз даних SQL і NoSQL для потреб проектів класу «розумне місто». Перш ніж передати його третій стороні, запропонований API забезпечив конфіденційність критичної інформації користувачів (CSP). У рамках запланованого MCTool запити перетворюються на відповідні моделі, а потім передаються до відповідної бази даних «розумного міста», враховуючи моделі, які вона може обробляти. Ключі шифрування та дешифрування метаданих гарантують, що мають лише авторизовані користувачі та адміністратор бази даних мати доступ до даних, що зберігаються в різних хмарах «розумного міста», і вносити зміни до них. Шифрування та дешифрування підтримуються запропонованою платформою в проектах класу «розумне місто».

3.2 Обговорення та класифікація особливостей використання різнотипових баз даних в проектах класу «розумне місто»

Дебати SQL проти NoSQL в проектах класу «розумне місто» не стосуються реляційних і нереляційних баз даних, незважаючи на назви. Моделі транзакцій обох баз даних аналізуються та порівнюються для потреб «розумних міст». Коли програма виконується, база даних виконує серію операцій, відомих як «транзакції». Усі транзакції в базі даних SQL залежать від властивостей ACID. У цьому випадку ACID відноситься до принципів атомізму, послідовності, ізоляції та довговічності. Однак розробники баз даних NoSQL дійшли висновку, що властивість ACID є марною перешкодою для захисту великих даних, особливо в проектах класу «розумне місто». Отже, на початку 2000-х років професор Ерік Брюер запропонував нову теорію. Принципами CAP є послідовність, доступність і толерантність до розділів. Теорема стверджує, що всі ці якості можуть бути

отримані одночасно дизайнерами, які працюють у розподіленому середовищі. В якості компромісу для гарантованої узгодженості та доступності розробники можуть реалізувати толерантну до розділів базу даних за допомогою моделі CA. Якщо розробник бази даних більше цінує доступність і толерантність до розділів, ніж узгодженість, то не можна сказати, що база даних заснована на AP. Щоб досягти узгодженості та толерантності до розділів без шкоди для доступності, розгортається база даних на основі CP. Таблиця 3.1 описує основні функції, які підтримуються обома різновидами баз даних.

Таблиця 3.1 – Можливості СУБД.

DBMSs	**Особливості**				
	Дані	**Схема**	**Відповідність масштабованості**	**Послідовність архітектури**	**Мова запитів**
RDBMS	S	Фіксована	Вертикальна ACID	Централізований Строгий	SQL
NoSQL	SUSm	Динамічна	Горизонтальна BASE	Розподілений Eventual	OO API, SQL Like

В таблиці позначено:

S – структурована.

SUSm – структуровані, неструктуровані та напівструктуровані.

LSWA – великомасштабні складські програми, дані давачів [81].

BFT – банківська справа, фінансові операції.

Стандартизація того, як класифікувати СУБД, призвела до кількох підкатегорій. Основою будь-якої системи керування базами даних є модель даних, на основі якої вона побудована. Багато сучасних комерційних СУБД значною мірою покладаються на реляційну модель даних. Модель об’єктних даних мало впроваджена, навіть якщо вона використовується в невеликій кількості комерційних систем. Багато застарілих програм продовжують

працювати в системах баз даних на основі ієрархічних і мережевих моделей даних. Можна класифікувати бізнес-модель NoSQL як одну з наступних.

Проблеми з надійністю та доступністю (CA) – ця структура бази даних надає пріоритет узгодженості та доступності даних за допомогою підходу реплікації. Частини бази даних не піклуються про допуски розділів. Якщо вузли розділені, дані не будуть синхронізовані. Vertica, Greenplum і системи управління реляційними базами даних є прикладами цього типу баз даних.

Є проблеми з узгодженістю та толерантністю до розділів (CP), які потрібно виправити. Основна мета такої системи керування базою даних – забезпечити цілісність даних, які вона зберігає. Однак висока доступність більше не підтримується. Дані зберігаються в різних вузлах, і коли вузол виходить з ладу, це призводить до того, що дані, які забезпечують узгодженість між ними, стають недоступними. Він підтримує толерантність до розділів, блокуючи повторну синхронізацію даних. Hypertable, BigTable і HBase є лише кількома прикладами систем баз даних, які підтримують CP та можуть бути корисними для для потреб проектів класу «розумне місто».

У цьому типі бази даних забезпечення доступності даних і допуску до розділів (AP) є головним пріоритетом. Якщо зв’язок між вузлами зривається, це не впливає на статус окремого вузла. Після вирішення розділу дані повторно синхронізуються, але узгодженість не забезпечується. Цих принципів дотримуються бази даних Riak, CouchDB і KAI.

Коли один елемент бази даних виходить з ладу, але інші продовжують працювати, цей розділ бази даних вважається «загалом доступним». У разі збою вузла операція продовжиться шляхом реплікації даних серед інших вузлів.

Системи баз даних NoSQL отримали ряд ознак, які можна пояснити тим, що структура реалізації не включає звичайний SQL. Кожна з кількох різних типів баз даних NoSQL використовує унікальний набір процедур запитів. Розробники програмного забезпечення часто несуть відповідальність за належне проектування виконання запитів, а не залежать від декларативної мови запитів, яка поєднує запитання та забезпечує швидкі плани виконання запитів. Це тому, що декларативна мова запитів може об’єднувати запитання. Користувачі

системи відповідають за виконання додаткових завдань, таких як перевірка та інтерпретація зібраної інформації. Крім того, відповідальність за забезпечення узгодженості даних, реплікацію даних і доступність під час одночасних змін у спільних і реплікованих базах даних все більше і більше переходить до рук розробників. Поєднання масивних систем даних із базами даних NoSQL може мати декілька різних впливів на дизайн та архітектуру програмного забезпечення в проектах класу «розумне місто».

Брюер сформулював набір вимог до розподілених баз даних, використовуючи свій метод CAP [82]. Ці вимоги служать базовими. У випадку, коли є мережевий розділ «P» – у кластері, коли інформація втрачається випадковим чином між вузлами. Узгодженість «C» – представлення ідентичних даних усім користувачам має мати пріоритет над доступністю. «A» – підтвердження має бути отримано кожним клієнтом як успіх або невдача. Якщо немає розділу «P», фреймворк «розумного міста» надає перевагу затримці «L», а не узгодженості «C», як показано в PARCELS.

Незважаючи на це, коли є розділ «P», структура буде наголошувати на доступності «A», а не на узгодженості «C». У програмному стані дані можуть змінюватися з часом залежно від таких факторів, як рівень участі користувача. Корисність такої інформації також може погіршитися після закінчення певного періоду. Тому необхідно або оновити, або отримати дані, щоб інформація була корисною в системі.

Відповідно до остаточної узгодженості, після кожної зміни даних дані не стають узгодженими в усій системі, але вони стануть узгодженими з часом. Зазначається, що дані залишаться точними в осяжному майбутньому.

На даний час недостатньо покладатися виключно на структуровані дані, оскільки неструктурована природа астрономічних даних вимагає швидкого аналізу даних і методів вилучення інформації для потреб проектів класу «розумне місто». Можливості бази даних SQL обмежені в їх здатності ефективно обробляти різні форми великих даних. Можливості баз даних NoSQL дають змогу ефективно обробляти масивні набори даних в проектах класу «розумне місто». Бази даних NoSQL перевершують у сферах розширення ємності

зберігання, гнучкості схеми, масштабованості та доступу в реальному часі. Бази даних NoSQL дотримуються атрибута BASE. Замість пріоритету узгодженості та безпеки даних, бази даних NoSQL наголошують на покращенні ефективності читання та запису даних. Застосунок відповідає за забезпечення узгодженості даних. З цієї причини це чудовий варіант для обробки великих наборів даних в проектах класу «розумне місто». Крім того, до баз даних NoSQL не застосовуються обмеження на рівні даних, стовпців або таблиць, які використовуються в структурованих базах даних.

У результаті наукометричного пошуку [13] проаналізовано понад сто сорок попередніх досліджень, які порівнювали корисність, продуктивність і надійність баз даних SQL та баз даних NoSQL для потреб проектів класу «розумне місто». При цьому було врахувало великі обсяги даних «розумних міст». Згідно з цим, бази даних NoSQL пропонують більший потенціал для розширення [83]. Бази даних SQL краще підходять для транзакційних систем і використовують більше ресурсів для забезпечення цілісності та узгодженості даних «розумних міст», тоді як бази даних NoSQL краще обладнані для обробки величезних і різноманітних наборів даних «розумних міст» і використовують менше ресурсів для забезпечення цілісності та узгодженості даних.

З іншого боку, бази даних NoSQL [49] відрізняються в тому сенсі, що вони надають більший пріоритет доступності даних в проектах класу «розумне місто». Результати нашого тестування показують, що реляційні бази даних не можна ефективно замінити базами даних NoSQL в проектах класу «розумне місто». Оскільки обидві бази даних мають свої переваги та недоліки, база даних, яку міська організація вибере для використання, залежатиме від вимог, які є унікальними для цього бізнесу. Щоб навести лише одну ілюстрацію, бази даних NoSQL, які дають змогу використовувати модуль програмування MapReduce, краще підходять для паралельних обчислень [84] для потреб проектів класу «розумне місто», коли вони реалізовані в середовищі кластера.

Бази даних NoSQL побудовані на схемі, яка є більш рухливою та динамічною, на відміну від реляційних баз даних, які сильно залежать від попередньо встановленої структури даних, яку називають «схемою», таблична

форма [85]. Наприклад, щоб відстежувати дані громадян, потрібно використовувати поля StdRegNo, StdName і StdAddress. Під час роботи з реляційними базами даних перше, що має відбутися, це побудова схеми, яка задовольняє всім необхідним вимогам домену «розумного міста» та цілісності.

Після цього можна буде зберегти відповідні дані студента та дотримуватися необхідних обмежень. Розглядаючи розширення обсягу поточної бази даних шляхом включення двох нових стовпців, слід приділити цьому певну увагу. Ті, хто буде відповідати за внесення змін до існуючої схеми, також відповідатимуть за перенесення даних із попередньої схеми до нової. Маючи справу з великими наборами даних в проектах класу «розумне місто», процедура може стати непрактично повільною та довгою. Той факт, що бази даних NoSQL не оновлюються автоматично щоразу, коли в схему вносяться нові зміни, становить найважливішу проблему для гнучкої розробки програмного забезпечення [21] для потреб проектів класу «розумне місто». Під час роботи з базами даних NoSQL необов'язково мати заздалегідь визначену схему для внесення будь-яких змін.

Нормалізація даних для контролю викидів, реляційної схеми (атрибутів), обмежень домену «розумного міста», обмежень перевірки, унікальних обмежень і обмежень Not NULL, серед іншого, сприяє безпечній інтеграції даних у реляційні бази даних на відміну від баз даних NoSQL [20]. Реляційні бази даних забезпечують добре налагоджену систему безпеки та автентифікації для користувачів в проектах класу «розумне місто». Бази даних SQL перевершують бази даних NoSQL як з точки зору продуктивності, так і довговічності. SQL є стандартним інтерфейсом для всіх реляційних баз даних, однак бази даних NoSQL не використовують SQL. Натомість вони використовують інший інтерфейс.

Бази даних SQL і NoSQL використовують різноманітні моделі [86] для потреб проектів класу «розумне місто». Крім того, існує різниця в базовій моделі даних, що використовується кожним типом бази даних NoSQL [87]. Враховуючи поширеність кількох CSP, важко досягти переносимості даних в проектах класу «розумне місто». Швидке розширення комерційних баз даних створює ще одну

проблему для хмарного зберігання. Натомість AWS DBaaS [1] пропонує дещо обмежені параметри розширення сховища для хмарних баз даних Azure.

3.3 Прогалини між різнотиповими базами даних «розумних міст»

Реалізація складних систем розподілу в проектах класу «розумне місто» потрібна для досягнення значних рівнів як доступності, так і масштабованості. Крім того, сегментування та розділення відбуваються на базових рівнях архітектури програмного забезпечення «розумного міста», які називаються рівнем застосунків, рівнем кешування та рівнем внутрішнього сховища відповідно. Однак досягти високої масштабованості для потреб проектів класу «розумне місто» може бути складно через атомарну абстракцію, представлену типовою структурою, яка використовує SQL як золотий стандарт непроцедурної мови. Крім того, програмне забезпечення має бути «розумним», щоб вирішувати задачі, які виникають із копіями даних, і невідповідності, спричинені колізіями між змінами реплік, які відбуваються одночасно в проектах класу «розумне місто». Що стосується критеріїв якості, таких як масштабованість, послідовність, продуктивність і довговічність, кожен різновид бази даних NoSQL має власний унікальний набір недоліків і обмежень. Крім того, системний архітектор принципово зобов’язаний провести дослідження характеристик номінальної бази даних, щоб виконати вимоги, висунуті постачальником під час вибору відповідної бази даних в проектах класу «розумне місто». Розглянемо прогалини кожного різновиду NoSQL баз даних.

Якщо програмам проектів класу «розумне місто» просто потрібно зберігати й отримувати елементи даних, прозорі для СУБД і ідентифіковані за допомогою ключа, сховище ключ-значення може бути найкращим варіантом для цього. Навпаки, програмне забезпечення «розумного міста» аварійно завершує роботу, якщо воно намагається виконати запит до бази даних на основі значення атрибута, відмінного від ключа. Крім того, неможливо оновити або отримати лише одне поле зі сховища ключ-значення документа.

Якщо застосункам «розумного міста» потрібен детальний контроль над тим, які записи отримувати, які поля в записі змінювати та які поля отримувати на основі критеріїв, відмінних від первинного ключа, бази даних документів є чудовим варіантом. Сховища даних документів забезпечують більшу гнучкість запитів, ніж сховища ключ-значення.

Коли програмам «розумного міста» потрібно зберігати дані з сотнями чи тисячами полів, але в більшості запитів потрібно отримати доступ лише до підмножини цих полів, сховища даних із сімейства стовпців є ефективним рішенням. Такі сховища даних добре підходять для масивних наборів даних в проектах класу «розумне місто».

Графові бази даних ідеально підходять для випадків використання, які включають зберігання та аналіз даних про сутності «розумного міста» зі складними зв’язками одна з одною. У графовій базі даних важливість сутностей та їхніх зв’язків розглядається однаково.

Термін «великі дані» [88] часто використовується для позначення даних, отриманих із великомасштабних сенсорних мереж «розумного міста» або створеної користувачами інформації з соціальних мереж [89]. Однак, оскільки багато людей не мають необхідних навичок, вони не знають, як працювати з даними «розумного міста», щоб досягти бажаного результату для свого бізнесу. Це пояснюється тим, що значна кількість технологій, методів і дисциплін, що стосуються великих даних «розумного міста», є відносно недавніми розробками. З іншого боку, реляційні сховища даних «розумного міста» не в змозі обробити таку велику колекцію даних, що вимагає створення нових можливостей системи зберігання. Дані, що зберігаються в базі даних NoSQL, не потребують централізованої схеми. Завдяки більшій масштабованості, доступності даних, відмовостійкості та швидкій обробці величезних обсягів неструктурованих даних бази даних NoSQL підходять для обробки великих даних для потреб проектів класу «розумне місто». Є багато питань, які ще не вирішені через відмінності між реляційними базами даних і базами даних NoSQL.

Тому можемо зробити висновок, що NoSQL не є прийнятною альтернативою реляційним базам даних в проектах класу «розумне місто». На

додаток до цього, NoSQL є чудовим варіантом для неоднорідних великих даних «розумного міста». Кожна з цих областей має численні незаповнені дослідницькі пустоти. Простота, масштабованість і продуктивність проектів баз даних NoSQL є областями, які потребують значного простору для вивчення. На відміну від суворої структури даних реляційної бази даних «розумного міста», бази даних NoSQL використовують плоский файл або модель даних ключ/значення в табличній формі. Бази даних NoSQL, завдяки своїй горизонтальній масштабованості, добре підходять для роботи з великою кількістю та різноманітністю даних, які зараз використовуються в проектах класу «розумне місто». Однак це не стосується баз даних SQL, які можуть масштабуватися вертикально. При порівнянні NoSQL з реляційними базами даних масштабованість і продуктивність є двома найважливішими факторами для потреб проектів класу «розумне місто». Існує також дослідницьке питання без відповіді про те, як інтегрувати функції реляційних баз даних із функціями нереляційних баз даних в проектах класу «розумне місто». Для баз даних NoSQL також недостатньо вивчено проблему проектування простих схем [90].

Розробка гнучких фреймворків міграції даних [87] з реляційної бази даних до сховища даних NoSQL «розумного міста» є ще одним напрямом досліджень, який отримав підвищений інтерес в останні роки. Успіх передачі даних SQL-toNoSQL поставлено на карту. Це надзвичайно важливо, оскільки будь-якій компанії необхідно отримувати корисні відомості зі своїх власних даних, наприклад про використання системних ресурсів і оцінки продуктивності співробітників.

Крім того, NoSQL є найкращим варіантом порівняно з реляційними базами даних у хмарі через розподілену природу хмарної платформи «розумного міста». Для того, щоб зменшити зусилля користувачів хмари під час переміщення даних між багатьма хмарними платформами та здійснювати контроль над сумісністю даних і проблемами переносимості, сучасний рівень вимагає уніфікованих інфраструктур API [63] для потреб проектів класу «розумне місто».

3.4 Результати дослідження програмних архітектур SQL та NoSQL баз даних для потреб «розумних міст»

Результати дослідження програмних архітектур SQL та NoSQL баз даних для потреб «розумних міст» показують, що немає необхідності переходити від реляційних баз даних до баз даних NoSQL. Оскільки обидва типи баз даних мають переваги та недоліки, муніципалітет, міська установа або компанія може вибрати СУБД, яка найкраще відповідає її вимогам.

Організація може використовувати базу даних SQL, якщо вона надає пріоритет стандартизації та узгодженості даних «розумного міста». NoSQL слід використовувати, коли місто має велику кількість неструктурованих даних і доступність даних є високою вимогою. Для агрегації невеликих наборів даних «розумного міста» можна віддати перевагу реляційній базі даних перед базою даних NoSQL і навпаки для аналітики великих даних.

MapReduce найбільше підходить для використання в кластерах «розумного міста» завдяки своїй розподіленій природі. Хоча MapReduce працює повільно під час процесу агрегації, він більш ефективний у паралельних обчисленнях і розроблений для обробки величезних обсягів неструктурованих даних.

Бази даних NoSQL, завдяки своїй розсіяній та високомасштабованій природі, добре підходять для програм «розумного міста», які генерують об’ємні дані з різноманітними функціями. Коли залучаються геопросторові дані, масштабованість реляційних баз даних вища, ніж у баз даних NoSQL. Бази даних NoSQL мають швидший час відповіді на дані, ніж реляційні бази даних, особливо при обробці величезних обсягів геопросторових даних «розумного міста».

Сховища даних NoSQL пропонують альтернативу звичайним RDBMS, але ймовірно, що організації не можуть відразу вирішити, який з них використовувати. Оптимальна стратегія вибору відповідної бази даних NoSQL для потреб проектів класу «розумне місто» полягає у визначенні того, що вимагає програма, а система керування реляційною базою даних не може забезпечити. Якщо система керування реляційною базою даних (RDBMS) може ефективно керувати даними «розумного міста», система зберігання NoSQL може

не знадобитися. Крім того, загальновизнано, що база даних NoSQL є новою розробкою в просторі баз даних. Однак вони розробляються з використанням загальновизнаних теорій. Незважаючи на переваги, системи на основі NoSQL не позбавлені недоліків, таких як відсутність загальноприйнятих стандартів або добре відомої мови запитів для використання з базами даних NoSQL. Кожна база даних має свої особливості та способи роботи. Навпаки, ці набори даних є новими та динамічними. Немає гарантії, що всі дані «розумного міста» будуть правильно записані в сховище даних, оскільки бази даних NoSQL не пропонують строгих характеристик ACID.

Крім того, бази даних NoSQL спрощують швидку розробку застосунків в проектах класу «розумне місто» завдяки своїй динамічній та гнучкій схемі. На відміну від жорсткої структури реляційних баз даних, моделі та архітектури баз даних NoSQL є більш адаптивними.Перехід від однієї моделі зберігання до іншої є складним для розробників через різні моделі, які використовуються базами даних NoSQL. Різні CSP використовують несумісні протоколи та інтерфейси, що робить їх несумісними один з одним. Оскільки кожен CSP розробив власні API для своїх конкретних служб, стає важко обмінюватися даними між ними для потреб проектів класу «розумне місто». Для ефективного керування портативністю та сумісністю даних бази даних NoSQL потребують єдиного стандартизованого хмарного рішення.

3.5 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи описано ефективність DBaaS, взаємодія та переносимість даних у різних NoSQL базах даних «розумних міст». Проведено обговорення та класифікація особливостей використання різнотипових баз даних в проектах класу «розумне місто». Висвітлено прогалини між різнотиповими базами даних «розумних міст». Подано результати дослідження програмних архітектур SQL та NoSQL баз даних для потреб «розумних міст».

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Долікарська допомога при контузіях

Кваліфікаційна робота освітнього рівня «магістр» присв’ячена дослідженню програмних архітектур SQL та NoSQL баз даних для потреб «розумних міст». Вітчизняні проекти «розумних міст» супроводжуються регулярними атаками з використанням ракетного озброєння та безпілотних літальних апаратів. Це супроводжується вибухами, які супроводжуються контузіями громадян.

Контузія може виникнути внаслідок ударної хвилі від вибуху або вдару особи об воду чи землю. Цей вид травми може супроводжуватися іншими ушкодженнями, такими як забій головного мозку, розрив барабанної перетинки, ушкодження легень або інших внутрішніх органів [91]. Залежно від ступеня тяжкості визначають три рівні контуження:

1. Легке контуження: симптоми включають заїкання, тремтіння чи поколювання кінцівок, погіршення слуху.

2. Середнього ступеня тяжкості: проявляється погіршенням зору, відсутністю реакції зіниць на світло, повною глухотою, частковим паралічем кінцівок.

3. Важке контуження: супроводжується втратою свідомості, переривчастим диханням, судомами чи мимовільними рухами кінцівками, може спостерігатися кровотеча з рота і вух.

Легке контуження, зазвичай, не має негативних наслідків для організму. Середнє та важке контуження може залишати серйозні наслідки, тому рятівникам слід готуватися до тривалого процесу реабілітації для постраждалого. В усіх випадках негайно потрібно викликати швидку допомогу.

Перед прибуттям медичної допомоги слід виконати такі кроки:

– Покинути небезпечну зону, якщо контуження виникло внаслідок вибуху.

– Розстебнути одяг постраждалого, який може обмежувати дихання або рух.

– Повернути постраждалу особу набік, щоб уникнути потрапляння блювотних мас в дихальні шляхи.

– Очистити носову та ротову порожнини від води, пилу, піску чи осколків вибуху.

– Якщо постраждалий не дихає чи має проблеми з диханням, слід здійснювати штучне дихання. Непрямий масаж серця не рекомендується, оскільки може завдати шкоди внутрішнім органам. Ритмічні натискання в область грудної клітини можуть лише погіршити стан потерпілого.

Додатково до перерахованих дій під час надання допомоги при контуженні важливо враховувати, що:

– Підтримка дихання та кровообігу: Забезпечте постраждалому постійний доступ до свіжого повітря. Якщо особа втратила свідомість і не дихає, починайте надавати штучне дихання та проводьте методи відновлення серцево-судинної системи. Використовуйте методи реанімації.

– Зупинка кровотеч: Якщо спостерігається кровотеча, застосовуйте необхідні заходи для її зупинки. Використовуйте стерильні пов'язки чи тампони, накладайте тиск на поранення та уникайте видалення забинтованих матеріалів, якщо кровотеча велика.

– Збереження тепла та комфорту: Постарайтеся забезпечити тепло та комфорт постраждалому, особливо якщо подія сталася в холодний чи вологий погодний день. Використовуйте одяг чи покривало для утеплення.

– Психологічна підтримка: Надайте постраждалому психологічну підтримку. Спілкуйтеся з ним, спробуйте зняти стрес та відвести його думки від травмуючої ситуації.

– Збереження умов безпеки: Намагайтеся утримувати навколишній простір вільним від небезпек та уникайте додаткових ризикованих ситуацій.

Зазначте, що це лише загальні рекомендації, і важливо враховувати конкретні умови та особливості події при наданні допомоги. Якщо можливо, завжди викликайте медичну допомогу та дотримуйтеся інструкцій фахівців.

4.2 Фактори що впливають на функціональний стан користувача комп'ю тера

В даній кваліфікаційній роботі освітнього рівня «магістр» проводиться дослідження програмних архітектур SQL та NoSQL баз даних для потреб «розумних міст». Інформаційно-технологічні платформи «розумних міст» впроваджуються з використанням різнотипової комп’ютерної техніки.

Широке впровадження комп’ютерної техніки в усі сфери життя людини призвело до як позитивних, так і негативних наслідків. Серед негативних наслідків – погіршення здоров’я користувачів комп’ютерів (КК).

Основною причиною негативного впливу комп’ютерів на здоров’я є недотримання принципів ергономіки, вимог санітарії та режиму роботи. Це може призвести до виникнення "ергономічних" захворювань, які пов’язані, зокрема, з впливом комп’ютерів та периферійних пристроїв на організм КК [92].

За даними Всесвітньої організації охорони здоров’я (ВООЗ), у користувачів ПК виявлено такі нові види захворювань:

– Синдром "комп’ютерного стресу оператора" – це стан підвищеної нервозності, тривожності, втоми, що виникає внаслідок тривалої роботи за комп’ютером.

– Травми повторних навантажень – це захворювання опорно-рухового апарату, що виникають внаслідок тривалого виконання одноманітних рухів.

– Фотоепілептичні приступи – це напади судом, що виникають внаслідок впливу світлових спалахів з певною частотою.

Організм людини в цілому не є байдужим до роботи з комп’ютерами. Найбільш вразливими виявляються зір, центральна нервова і кістково-м’язова системи організму КК.

Щоб знизити негативний вплив комп’ютерів на здоров’я, необхідно дотримуватися таких рекомендацій:

Влаштовувати регулярні перерви в роботі за комп’ютером.

– Правильно організувати робоче місце.

– Використовувати ергономічні засоби захисту зору.

– Дотримуватися режиму дня та відпочинку.

Людський організм чутливий до впливу комп’ютерів. Зокрема, найбільше страждають зір, центральна нервова та кістково-м’язова системи. Це підтверджують дослідження Всесвітньої організації охорони здоров’я. Зокрема, у висновках експертів ВООЗ, розроблених на основі проведених у різних державах світу досліджень, чітко визначено, що:

– Найбільше навантаження під час роботи за ПК припадає на зоровий аналізатор;

– Робота із засобами обчислювальної техніки є стресовим фактором для користувача;

– Людина, яка працює з комп’ютерними засобами, зазнає впливу фізичних факторів різної природи й малої інтенсивності, а про деякі з них поки що немає достатніх наукових даних, щоб визначити рівень їх впливу на здоров’я людини.

Функціональний стан та здоров’я молодої людини, яка працює за комп’ютером, залежить від багатьох факторів. Це інтенсивність та тривалість роботи, якість програмного забезпечення, його ергономічні, педагогічні та психогігієнічні властивості, а також рівень "дружності" інтерфейсу. Крім того, важливу роль відіграють об’єктивні фактори внутрішнього середовища, такі як мікроклімат, освітленість, яскравість, контрастність і колір зображення на екрані, іонізуюче та неіонізуюче опромінення, шум тощо.

Загалом, прийнято виділяти чотири групи основних об’єктивних факторів, які можуть негативно вплинути на здоров’я будь-якого користувача персонального комп’ютера:

– Візуальні параметри дисплеїв ПК у сполученні зі світловим кліматом у робочому приміщенні (комп’ютерному класі).

– Електростатичне і електромагнітне поля комп’ютера, дисплея та інших периферійних пристроїв (емісійні параметри).

– Ергономічні параметри робочого місця та приміщення.

– Режим праці й відпочинку, види й напруженість роботи за комп’ютером.

Розглянемо фактори, які впливають на здоров’я користувачів комп’ютерів (КК), та заходи щодо зменшення їх негативного впливу.

Зір. Користувачі комп’ютерів, які щоденно працюють за ПК, ризикують захворіти на комп’ютерний зоровий синдром (CVS). Ознаками CVS є такі симптоми:

– головні болі;

– напруженість очей;

– двоїння зображення;

– стомлені, червоні або сухі очі;

– тимчасова короткозорість;

– випадкове “змазування” зображень на екрані;

– зростаюче подразнення очей;

– зміни колірного сприйняття.

Щоб зменшити ризик виникнення CVS, слід дотримуватися таких рекомендацій:

– Розміщуйте екран комп’ютера на відстані 50-70 см від очей.

– Налаштуйте яскравість та контрастність екрана так, щоб вони відповідали освітленості в приміщенні.

– Регулярно робіть перерви в роботі за комп’ютером, під час яких дивіться в далечінь.

– Використовуйте захисні окуляри для комп’ютерів.

Психіка. Вплив комп’ютерів на психіку людини є найменш вивченим питанням. Однак існують численні факти погіршення психічного здоров’я КК, зокрема, Інтернет-залежність та комп’ютерна ігроманія.

Щоб знизити ризик виникнення проблем із психікою, слід дотримуватися таких рекомендацій:

– Обмежувати час роботи за комп’ютером.

– Не використовувати комп’ютер для вирішення проблем, пов’язаних із емоційним станом.

– Брати участь у соціальній діяльності та підтримувати зв’язки з близькими людьми.

Тривале використання комп’ютерів може призвести до погіршення зору, а також до нервово-психічних розладів, таких як головні болі, спазми

мускулатури обличчя та інші. У дітей це може призвести до виникнення нового захворювання – синдрому відеоігрової епілепсії.

Щоб зменшити ризик виникнення цих проблем, слід дотримуватися таких рекомендацій:

– Обмежувати час роботи за комп’ютером.

– Робити регулярні перерви в роботі за комп’ютером.

– Розміщувати екран комп’ютера на відстані 50-70 см від очей.

– Налаштувати яскравість та контрастність екрана так, щоб вони відповідали освітленості в приміщенні.

– Використовувати захисні окуляри для комп’ютерів.

Комп’ютери, дисплеї та інші периферійні пристрої генерують електромагнітні поля. Вплив цих полів на здоров’я людини досі повністю не вивчений, але є дані, які свідчать про потенційну небезпеку для здоров’я, яку може спричинити довготривале перебування у зоні неіонізованих електромагнітних полів вкрай низьких частот (5‚2000 Гц) та дуже низьких частот (2‚400 кГц).

Серед усіх пристроїв, що входять у стандартну комплектацію персонального комп’ютера, найбільш "шкідливим" є монітор. Монітор є джерелом різного виду випромінювань, зокрема м'якого рентгенівського, оптичного ультрафіолетового, інфрачервоного, радіочастотного та низькочастотного діапазонів електромагнітних і електростатичних полів.

Щоб зменшити ризик виникнення проблем, пов’язаних з впливом електромагнітних полів, слід дотримуватися таких рекомендацій:

– Розміщуйте комп’ютер на відстані не менше 1 метра від стіни.

– Не розміщуйте комп’ютер поруч з ліжком або диваном, де ви спите.

– Регулярно провітрюйте приміщення, де ви працюєте за комп’ютером.

Основним джерелом несприятливого впливу монітора є створювана ним напруженість електромагнітного поля [93]. Статичні і низькочастотні електромагнітні поля можуть призвести до таких проблем зі здоров’ям:

– захворювання шкіри (вугреві висипи, себорроїдна екзема, рожевий лишай та ін.);

– хвороби серцевосудинної системи;

– хвороби кишково-шлункового тракту;

– порушення роботи імунної системи;

– пухлини, зокрема, злоякісні.

Щоб зменшити ризик виникнення цих проблем, слід дотримуватися таких рекомендацій:

– розміщуйте комп’ютер на відстані не менше 1 метра від стіни;

– не розміщуйте комп’ютер поруч з ліжком або диваном, де ви спите;

– регулярно провітрюйте приміщення, де ви працюєте за комп’ютером;

– використовуйте монітор з низьким рівнем випромінювання.

Вплив мікроклімату. Робоче приміщення (кабінет), обладнаний комп’ютерною технікою, має відповідати таким вимогам:

– розміщуватися в окремій кімнаті із природним освітленням та організованим обміном повітря;

– бути досить просторим, ясним, тихим;

– мати сприятливі умови мікроклімату в усі пори року.

Вплив характеру, тривалості та інтенсивності роботи. Характер, тривалість та інтенсивність роботи за комп’ютером, режим праці і відпочинку є визначальними факторами впливу на здоров’я користувача комп’ютера.

Для всіх користувачів комп’ютерів важливо дотримуватися таких рекомендацій:

– обмежувати час роботи за комп’ютером;

– робити регулярні перерви в роботі за комп’ютером;

– дотримуватися правил ергономіки при роботі за комп’ютером.

Для дітей, школярів і студентської молоді важливо також враховувати такі рекомендації:

– тривалість роботи за комп’ютером не повинна перевищувати 20 хвилин на кожну годину занять;

– кожен навчальний день слід починати і закінчувати не з роботи за комп’ютером;

– щодня необхідно проводити хоча б одну годину на свіжому повітрі.

Вплив роботи за комп’ютером на здоров’я. Праця за комп’ютером може негативно впливати на здоров’я, викликаючи такі проблеми [58]:

– погіршення функціонального стану центральної нервової системи;

– значну втому зорового аналізатора;

– погіршення психологічного стану і працездатності.

Ці проблеми викликані такими факторами:

– специфічними для операторів ПК завданнями;

– високою відповідальністю за кінцевий результат;

– великим зоровим і нервово-емоційним напруженням.

Раціональний режим роботи за комп’ютером. Щоб зменшити негативний вплив роботи за комп’ютером на здоров’я, слід дотримуватися таких рекомендацій:

– обмежувати час роботи за комп’ютером;

– робити регулярні перерви в роботі за комп’ютером;

– дотримуватися правил ергономіки при роботі за комп’ютером.

Для викладачів, які працюють за комп’ютером, рекомендовано дотримуватися такого режиму роботи:

– загальна тривалість роботи у комп’ютерному класі впродовж дня не повинна перевищувати шести годин;

– тривалість безупинної роботи за комп’ютером має не перевищувати двох годин, після чого необхідна перерва тривалістю 15-20 хвилин.

Вплив роботи за комп’ютером на здоров’я студентів. Тривалість роботи за комп’ютером для студентів має бути обмежена. Для студентів 1-2 курсів вона не повинна перевищувати двох годин на день, а для студентів старших курсів – трьох годин. При цьому робота за відеотерміналом не повинна становити більше 50% від усього часу роботи з використанням комп’ютера.

Щоб зменшити негативний вплив роботи за комп’ютером на здоров’я студентів, слід дотримуватися таких рекомендацій:

– робити регулярні перерви в роботі за комп’ютером. Через кожні 20-25 хвилин занять слід робити паузи для виконання вправ для очей, а через 40-50

хвилин роботи варто влаштовувати 10-15-хвилинну перерву з фізкультурними вправами.

– дотримуватися правил ергономіки при роботі за комп’ютером.

Загальні рекомендації щодо зменшення негативного впливу роботи за комп’ютером на здоров’я. Комп’ютерна техніка може негативно впливати на здоров’я і фізичний стан людини. Щоб зменшити цей вплив, слід дотримуватися таких рекомендацій:

– облаштовувати та обладнати комп’ютерні кабінети відповідно до санітарних, ергономічних, гігієнічних норм.

– нормувати тривалість роботи за комп’ютером, зокрема, учнів (студентів) і вчителів (викладачів).

– проводити певні фізкультурно-оздоровчі заходи.

Дотримання цих рекомендацій дасть змогу усім працюючим за комп’ютерами значно зменшити їх вплив на здоров’я, фізичний стан та психіку людини.

4.3 Висновок до четвертого розділу

В четвертому розділі кваліфікаційної роботи описано долікарську допомогу при контузіях. Подано фактори що впливають на функціональний стан користувача комп'ю тера.

ВИСНОВКИ

В кваліфікаційній роботі освітнього рівня «Магістр» проведено дослідження програмних архітектур SQL та NoSQL баз даних для потреб «розумних міст».

В першому розділі кваліфікаційної роботи:

– Висвітлено актуальність дослідження предметної області використання SQL та NoSQL баз даних у «розумних містах».

– Описано стан проблемної області використання SQL та NoSQL баз даних для «розумних міст».

– Подано критерії пошуку наукових публікацій щодо використання SQL та NoSQL баз даних у «розумних містах».

– Розглянуто задіяні пошукові ресурси.

– Сформовано стратегію наукометричного пошуку матеріалів щодо використання SQL та NoSQL баз даних для «розумних міст».

– Описано процес і критерії відбору наукових публікацій.

– Подано критерії включення наукових публікацій.

– Наведено критерії виключення наукових публікацій.

– Розглянуто збір і вилучення даних щодо наукових публікацій.

В другому розділі кваліфікаційної роботи:

– Проведено аналіз і класифікацію результатів наукометричного пошуку.

– Подано результати пошуку наукових публікацій.

– Проаналізовано результати наукометричного пошуку.

– Виконано моделювання даних NoSQL MongoDB для потреб «розумних міст».

В третьому розділі кваліфікаційної роботи:

– Описано ефективність DBaaS, взаємодія та переносимість даних у різних NoSQL базах даних «розумних міст».

– Проведено обговорення та класифікація особливостей використання різнотипових баз даних в проектах класу «розумне місто».

– Висвітлено прогалини між різнотиповими базами даних «розумних міст».

– Подано результати дослідження програмних архітектур SQL та NoSQL баз даних для потреб «розумних міст».

В четвертому розділі кваліфікаційної роботи описано долікарську допомогу при контузіях. Подано фактори що впливають на функціональний стан користувача комп'ю тера.

ПЕРЕЛІК ДЖЕРЕЛ

1 Siddiqa, A.; Hashem, I.A.T.; Yaqoob, I.; Marjani, M.; Shamshirband, S.; Gani, A.; Nasaruddin, F. A survey of big data management: Taxonomy and state-of-the-art. J. Netw. Comput. Appl. 2016, 71, 151–166.

2 Kunanets N. et al. (2021) Designing the Repository of Documentary Cultural Heritage. In: Shakhovska N., Medykovskyy M.O. (eds) Advances in Intelligent Systems and Computing V. CSIT 2020. Advances in Intelligent Systems and Computing, vol 1293, pp 1034-1044. Springer, Cham. ISBN978-3-030-63270-0.

3 Kong, X.; Shi, Y.; Yu, S.; Liu, J.; Xia, F. Academic social networks: Modeling, analysis, mining and applications. J. Netw. Comput. Appl. 2019, 132, 86–103.

4 Ordonez, C. Optimization of Linear Recursive Queries in SQL. IEEE Trans. Knowl. Data Eng. 2009, 22, 264–277.

5 Obasanjo, D. Building scalable Databases: Denormalization, the NoSQL movement and Digg. 2009.

6 Strozzi, C. NoSQL—A Relational Database Management System. 2007–2010. Available online: http//www.strozzi.it/cgi-bin/ CSA/tw7/I/en_US/nosql/Home%20Page.

7 George, S. NoSQL—NOT ONLY SQL. Int. J. Enterp. Comput. Bus. Syst. 2013, 2.

8 Mansouri, Y.; Prokhorenko, V.; Babar, M.A. An automated implementation of hybrid cloud for performance evaluation of distributed databases. J. Netw. Comput. Appl. 2020, 167, 102740.

9 Kumari, A.; Tanwar, S.; Tyagi, S.; Kumar, N.; Parizi, R.M.; Choo, K.-K.R. Fog data analytics: A taxonomy and process model. J. Netw. Comput. Appl. 2019, 128, 90–104.

10 Bodnarchuk I., Duda O., Kharchenko A., Kunanets N., Matsiuk O., Pasichnyk V. Choice method of analytical information-technology platform for projects associated to the smart city class. ICTERI 2020 ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer

Proceedings of the 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume I: Main Conference p.317-330.

11 Kitchenham, B.; Charters, S. Guidelines for Performing Systematic Literature Reviews in Software Engineering; Keele University: Keele, UK, 2007.

12 Hosseinzadeh, S.; Rauti, S.; Laurén, S.; Mäkelä, J.-M.; Holvitie, J.; Hyrynsalmi, S.; Leppänen, V. Diversification and obfuscation techniques for software security: A systematic literature review. Inf. Softw. Technol. 2018, 104, 72–93.

13 Khan, Wisal, et al. "SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review." Big Data and Cognitive Computing 7.2 (2023): 97.

14 Petersen, K.; Vakkalanka, S.; Kuzniarz, L. Guidelines for conducting systematic mapping studies in software engineering: An update. Inf. Softw. Technol. 2015, 64, 1–18.

15 Badampudi, D.; Wohlin, C.; Petersen, K. Experiences from using snowballing and Database searches in systematic literature studies. In Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering, Nanjing, China, 27–29 April 2015; p. 17.

16 Ravi, K.; Khandelwal, Y.; Krishna, B.S.; Ravi, V. Analytics in/for cloud-an interdependence: A review. J. Netw. Comput. Appl. 2018, 102, 17–37.

17 Petersen, K.; Gencel, C. Worldviews, research methods, and their relationship to validity in empirical software engineering research. In Proceedings of the 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement, Ankara, Turkey, 23–26 October 2013; pp. 81–89.

18 Alsolai, H.; Roper, M. A systematic literature review of machine learning techniques for software maintainability prediction. Inf. Softw. Technol. 2020, 119, 106214.

19 Rodrigues, M.; Santos, M.Y.; Bernardino, J. Big data processing tools: An experimental performance evaluation. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 2019, 9, e1297.

20 Hou, B.; Qian, K.; Li, L.; Shi, Y.; Tao, L.; Liu, J. MongoDB NoSQL injection analysis and detection. In Proceedings of the 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud), Beijing, China, 25–27 June 2016; pp. 75–78.

21 Padhy, R.P.; Patra, M.R.; Satapathy, S.C. RDBMS to NoSQL: Reviewing some next-generation non-relational Database's. Int. J. Adv. Eng. Sci. Technol. 2011, 11, 15–30.

22 El Mouden, Z.A.; Jakimi, A.; Hajar, M.; Boutahar, M. Graph Schema Storage in SQL Object-Relational Database and NoSQL Document-Oriented Database: A Comparative Study. In Proceedings of the International Conference Europe Middle East & North Africa Information Systems and Technologies to Support Learning, Marrakech, Morocco, 21–23 November 2019; pp. 176–183.

23 Rathika, V. Graph-Based Denormalization for Migrating Big Data from SQL Database to NoSQL Database. In Proceedings of the Intelligent Communication Technologies and Virtual Mobile Networks, Tirunelveli, India, 14–15 February 2019; pp. 546–556.

24 Jung, M.-G.; Youn, S.-A.; Bae, J.; Choi, Y.-L. A study on data input and output performance comparison of MongoDB and PostgreSQL in the big data environment. In Proceedings of the 2015 8th International Conference on Database Theory and Application (DTA), Jeju Island, Republic of Korea, 28–25 November 2015; pp. 14–17.

25 Fatima, H.; Wasnik, K. Comparison of SQL, NoSQL and NewSQL Databases for internet of things. In Proceedings of the 2016 IEEE Bombay Section Symposium (IBSS), Maharashtra, India, 21–22 December 2016; pp. 1–6.

26 Wei-Ping, Z.; Ming-Xin, L.I.; Huan, C. Using MongoDB to implement textbook management system instead of MySQL. In Proceedings of the 2011 IEEE 3rd International Conference on Communication Software and Networks, Xian, China, 27–29 May 2011; pp. 303–305.

27 Duda, O., et al, Selection of Effective Methods of Big Data Analytical Processing in Information Systems of Smart Cities. CEUR Workshop Proceedings 2631, pp. 68-78. 2020.

28 Ray, P.P.; Dash, D.; De, D. Edge computing for Internet of Things: A survey, e-healthcare case study and future direction. J. Netw. Comput. Appl. 2019, 140, 1–22.

29 Aghi, R.; Mehta, S.; Chauhan, R.; Chaudhary, S.; Bohra, N. A comprehensive comparison of SQL and MongoDB Databases. Int. J. Sci. Res. Publ. 2015, 5, 1–

30 Faraj, A.; Rashid, B.; Shareef, T. Comparative study of relational and non-relations Database performances using Oracle and MongoDB systems. J. Impact Factor 2014, 5, 11–22.

31 Singh, A. Data Migration from Relational Database to MongoDB. Glob. J. Comput. Sci. Technol. 2019.

32 Pereira, D.A.; de Morais, W.O.; Pignaton de Freitas, E. NoSQL real-time Database performance comparison. Int. J. Parallel Emergent Distrib. Syst. 2018, 33, 144–156.

33 Duda, O., Pasichnyk, V., Kunanets, N., Antonii, R., Matsiuk, O. Multidimensional Representation of COVID-19 Data Using OLAP Information Technology. International Scientific and Technical Conference on Computer Sciences and Information Technologies, 2020, 2, pp. 277–280, 9321889.

34 Miranskyy, A.V.; Al-zanbouri, Z.; Godwin, D.; Bener, A.B. Database engines: Evolution of greenness. J. Softw. Evol. Process. 2017, 30, e1915.

35 Stanescu, L.; Brezovan, M.; Burdescu, D.D. Automatic mapping of MySQL Databases to NoSQL MongoDB. In Proceedings of the 2016 Federated Conference on Computer Science and Information Systems (FedCSIS), Gdansk, Poland, 11–14 September 2016; pp. 837–840.

36 Chapple, M. The Acid Model. Available online: http//Databases.about.com/od/specificproducts/a/acid.htm.

37 Chandra, D.G. BASE analysis of NoSQL database. Futur. Gener. Comput. Syst. 2015, 52, 13–21.

38 Zhao, G.; Huang, W.; Liang, S.; Tang, Y. Modeling MongoDB with relational model. In Proceedings of the 2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies, Washington, DC, USA, 9–11 September 2013; pp. 115–121.

39 Gonzalez-Aparicio, M.T.; Younas, M.; Tuya, J.; Casado, R. Evaluation of ACE properties of traditional SQL and NoSQL big data systems. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, Limassol, Cyprus, 8–12 April 2019; pp. 1988–1995.

40 Lee, C.-H.; Zheng, Y.-L. Automatic SQL-to-NoSQL schema transformation over the MySQL and HBase Databases. In Proceedings of the 2015 IEEE International Conference on Consumer Electronics-Taiwan, Taipei, Taiwan, 6–8 June 2015; pp. 426–427.

41 Kuzochkina, A.; Shirokopetleva, M.; Dudar, Z. Analyzing and Comparison of NoSQL DBMS. In Proceedings of the 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkov, Ukraine, 9–12 October 2018; pp. 560–564.

42 Anand, V.; Rao, C.M. MongoDB and Oracle NoSQL: A technical critique for design decisions. In Proceedings of the 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), Pudukkottai, India, 24–16 April 2016; pp. 1–4.

43 Simanjuntak, H.T.A.; Simanjuntak, L.; Situmorang, G.; Saragih, A. Query Response Time Comparison NOSQLDB MONGODB with SQLDB Oracle. JUTI J. Ilm. Teknol. Inf. 2015, 13, 95–105.

44 Dean, J.; Ghemawat, S. MapReduce: Simplified data processing on large clusters. Commun. ACM 2008, 51, 107–113.

45 Pokorný, J. Database technologies in the world of big data. In Proceedings of the 16th International Conference on Computer Systems and Technologies, Dublin, Ireland, 25–26 June 2015; pp. 1–12.

46 Yang, H.; Dasdan, A.; Hsiao, R.-L.; Parker, D.S. Map-reduce-merge: Simplified relational data processing on large clusters. In Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, Portland, OR, USA, 14–19 June 2007; pp. 1029–1040.

47 Nykiel, T.; Potamias, M.; Mishra, C.; Kollios, G.; Koudas, N. MRShare: Sharing across multiple queries in MapReduce. Proc. VLDB Endow. 2010, 3, 494–505.

48 Moniruzzaman, A.B.M.; Hossain, S.A. Nosql Database: New era of Databases for big data analytics-classification, characteristics and comparison. arXiv 2013, arXiv:1307.0191.

49 Zeng, N.; Zhang, G.-Q.; Li, X.; Cui, L. Evaluation of relational and NoSQL approaches for patient cohort identification from heterogeneous data sources. In Proceedings of the 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Kansas City, MO, USA, 13–16 November 2017; pp. 1135–1140.

50 Băzăr, C.; Iosif, C.S. The transition from rdbms to nosql. a comparative analysis of three popular non-relational solutions: Cassandra, mongodb and couchbase. Database Syst. J. 2014, 5, 49–59.

51 Burzan´ska, M.; Wis´niewski, P. How Poor Is the 'Poor Man's Search Engine'? In Proceedings of the International Conference: Beyond Databases, Architectures and Structures, Poznan´, Poland, 18–20 September 2018; pp. 294–305.

52 Bartoszewski, D.; Piorkowski, A.; Lupa, M. The comparison of processing efficiency of spatial data for PostGIS and MongoDB databases. In Proceedings of the Beyond Databases, Architectures and Structures. Paving the Road to Smart Data Processing and Analysis: 15th International Conference, BDAS 2019, Ustron´, Poland, 28–31 May 2019; pp. 291–302.

53 Fraczek, K.; Plechawska-Wojcik, M. Comparative analysis of relational and non-relational Databases in the context of performance in web applications. In Proceedings of the International Conference: Beyond Databases, Architectures and Structures, Ustron´, Poland, 30 May–2 June 2017; pp. 153–164.

54 Longley, P.A.; Goodchild, M.F.; Maguire, D.J.; Rhind, D.W. Geographic Information Systems and Science; John Wiley & Sons: Hoboken, NJ, USA, 2005.

55 Chromiak, M.; Stencel, K. A data model for heterogeneous data integration architecture. In Proceedings of the International Conference: Beyond Databases, Architectures and Structures, Ustron, Poland, 27–30 May 2014; pp. 547–556.

56 Yue, P.; Tan, Z. 1.06 GIS Databases and NoSQL Databases. Compr. Geogr. Inf. Syst. 2017, 50.

57 Piórkowski, A. Mysql spatial and postgis–implementations of spatial data standards. EJPAU 2011, 14, 3.

58 Kothuri, R.; Godfrind, A.; Beinat, E. Pro Oracle Spatial for Oracle Database 11g; Dreamtech Press: New Delhi, India, 2008.

59 Baralis, E.; Dalla Valle, A.; Garza, P.; Rossi, C.; Scullino, F. SQL versus NoSQL Databases for geospatial applications. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 3388–3397.

60 Roy-Hubara, N.; Sturm, A. Exploring the Design Needs for the New Database Era. In Enterprise, Business-Process and Information Systems Modeling; Springer: Cham, Switzerland, 2018; pp. 276–290.

61 Mehmood, E.; Anees. T. Performance Analysis of Not Only SQL Semi-Stream Join Using MongoDB for Real-Time Data Warehousing. IEEE Access 2019, 7, 134215–134225.

62 Okman, L.; Gal-Oz, N.; Gonen, Y.; Gudes, E.; Abramov, J. Security issues in nosql databases. In Proceedings of the 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, Changsha, China, 16 November 2011; pp. 541–547.

63 Alomari, E.; Noaman, A. SeCloudDB: A Unified API for Secure SQL and NoSQL Cloud Databases. In Proceedings of the 2019 3rd International Conference on Cloud and Big Data Computing, Oxford, UK, 28–30 August 2019; pp. 38–42.

64 Stravoskoufos, K.; Preventis, A.; Sotiriadis, S.; Petrakis, E.G.M. A Survey on Approaches for Interoperability and Portability of Cloud Computing Services. In CLOSER; Technical University of Crete: Chania, Greece, 2014; pp. 112–117.

65 Shirazi, M.N.; Kuan, H.C.; Dolatabadi, H. Design patterns to enable data portability between clouds' Databases. In Proceedings of the 2012 12th International Conference on Computational Science and Its Applications, Salvador, Bahia, 18–21 June 2012; pp. 117–120.

66 Alomari, E.; Barnawi, A.; Sakr, S. Cdport: A framework of data portability in cloud platforms. In Proceedings of the 16th International Conference on Information

Integration and Web-based Applications & Services, Singapore, 28–30 November 2014; pp. 126–133.

67 Vanelli, B.; da Silva, M.P.; Manerichi, G.; Pinto, A.S.R.; Dantas, M.A.R.; Ferrandin, M.; Boava, A. Internet of Things Data Storage Infrastructure in the Cloud Using NoSQL Databases. IEEE Lat. Am. Trans. 2017, 15, 737–743.

68 Grolinger, K.; Higashino, W.A.; Tiwari, A.; Capretz, M.A. Data management in cloud environments: NoSQL and NewSQL data stores. J. Cloud Comput. Adv. Syst. Appl. 2013, 2, 22.

69 Marston, S.; Li, Z.; Bandyopadhyay, S.; Zhang, J.; Ghalsasi, A. Cloud computing—The business perspective. Decis. Support Syst. 2011, 51, 176–189.

70 Kolev, B.; Bondiombouy, C.; Valduriez, P.; Jiménez-Peris, R.; Pau, R.; Pereira, J. The cloudmdsql multistore system. In Proceedings of the 2016 International Conference on Management of Data, San Francisco, CA, USA, 26 June–1 July 2016; pp. 2113–2116.

71 Loutas, N.; Kamateri, E.; Tarabanis, K. A semantic Interoperability framework for cloud platform as a service. In Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science, Washington, DC, USA, 29 November–1 December 2011; pp. 280–287.

72 Zhou, L.; Fu, A.; Yu, S.; Su, M.; Kuang, B. Data integrity verification of the outsourced big data in the cloud environment: A survey. J. Netw. Comput. Appl. 2018, 122, 1–15.

73 Loutas, N.; Kamateri, E.; Bosi, F.; Tarabanis, K. Cloud computing Interoperability: The state of play. In Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science, Washington, DC, USA, 29 November–1 December 2011; pp. 752–757.

74 Escalera, M.F.P.; Chavez, M.A.L. UML model of a standard API for cloud computing application development. In Proceedings of the 2012 9th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, Mexico, 26–28 September 2012; pp. 1–8.

75 Liao, C.-S.; Shih, J.-M.; Chang, R.-S. Simplifying MapReduce data processing. Int. J. Comput. Sci. Eng. 2013, 8, 219–226.

76 Curino, C.; Jones, E.P.; Popa, R.A.; Malviya, N.; Wu, E.; Madden, S.; Balakrishnan, H.; Zeldovich, N. Relational Cloud: A Database-as-a-Service for the Cloud; MIT Libraries: Boston, MA, USA, 2011.

77 Indu, I.; PM, R.A.; Bhaskar, V. Encrypted token based authentication with adapted SAML technology for cloud web services. J. Netw. Comput. Appl. 2017, 99, 131–145.

78 Sakr, S.; Liu, A.; Batista, D.M.; Alomari, M. A Survey of Large Scale Data Management Approaches in Cloud Environments. IEEE Commun. Surv. Tutor. 2011, 13, 311–336.

79 Lehner, W.; Sattler, K.-U. Database as a service (DBaaS). In Proceedings of the 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010), Long Beach, CA, USA, 1–6 March 2010; pp. 1216–1217.

80 Kiefer, T.; Lehner, W. Private table Database virtualization for dbaas. In Proceedings of the 2011 Fourth IEEE International Conference on Utility and Cloud Computing, Melbourne, Austalia, 5–8 December 2011; pp. 328–329.

81 Duda O., Matsiuk O., Kunanets N., Pasichnyk V., Rzheuskyi A., Bilak Y., Formation of Hypercubes Based on Data Obtained from Systems of IoT Devices of Urban Resource Networks, International Journal of Sensors, Wireless Communications and Control (2020) 10: 1. ISSN 2210-3287.

82 Brewer, E.A. Towards robust distributed systems. In PODC; Inktomi: Foster City, CA, USA, 2000; Volume 7.

83 Sahatqija, K.; Ajdari, J.; Zenuni, X.; Raufi, B.; Ismaili, F. Comparison between relational and NOSQL Databases. In Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 21–25 May 2018; pp. 216–221.

84 Alias, N.; Suhari, N.N.; Saipol, H.F.; Dahawi, A.A.; Saidi, M.M.; Hamlan, H.A.; Teh, C.R. Parallel computing of numerical schemes and big data analytic for solving real life applications. Jurnal Teknologi. 2016, 78.

85 Chang, M.-L.E.; Chua, H.N. SQL and NoSQL Database Comparison. In Proceedings of the Future of Information and Communication Conference, Vienna, Austria, 4–6 December 2018; pp. 294–310.

86 Lian, J.; Miao, S.; McGuire, M.; Tang, Z. SQL or NoSQL? In Which Is the Best Choice for Storing Big Spatio-Temporal Climate Data? In Proceedings of the International Conference on Conceptual Modeling, Xi'an, China, 22–25 October 2018; pp. 275–284.

87 Lee, C.-H.; Zheng, Y.-L. SQL-to-NoSQL schema denormalization and migration: A study on content management systems. In Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics, Hong Kong, China, 9–12 October 2015; pp. 2022–2026.

88 Frizzo-Barker, J.; Chow-White, P.A.; Mozafari, M.; Ha, D. An empirical study of the rise of big data in business scholarship. Int. J. Inf. Manag. 2016, 36, 403–413.

89 Zafar, R.; Yafi, E.; Zuhairi, M.F.; Dao, H. Big data: The NoSQL and RDBMS review. In Proceedings of the 2016 International Conference on Information and Communication Technology (ICICTM), Kuala Lumpur, Malaysia, 16–17 May 2016; pp. 120–126.

90 Kumar, M.S. Comparison of NoSQL Database and Traditional Database-An emphatic analysis. JOIV Int. J. Inform. Vis. 2018, 2, 51–55.

91 Редакція Т. Контузія - симптоми та перша допомога. TCH.ua. URL: https://tsn.ua/ukrayina/kontuziya-simptomi-ta-persha-dopomoga-2022373.html.

92 Вплив комп'ютера на здоров'я користувача http://mirgorod-gorono.at.ua/publ/metodob_39_ednannja_vchiteliv/inform/vpliv_komp_jutera_na_zdorov_ja_koristuvacha/26-1-0-238.

93 Фактори впливу на функціональний стан користувачів комп'ютерів https://infopedia.su/15xefda.html.

ДОДАТКИ

Додаток А

Тези конференції

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

XI НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

13-14 грудня 2023 року

ТЕРНОПІЛЬ
2023

УДК 004.9

Лебідко Д.М., Онуферко В.А., Перетятко Т.П.
Тернопільський національний технічний університет імені Івана Пулюя, Україна

ВЕЛИКІ ЗА ОБСЯГОМ ДАНІ, РЕЛЯЦІЙНІ ТА НЕРЕЛЯЦІЙНІ МОДЕЛІ

Lebidko D.M., Onuferko V.A., Peretiatko T.P.

BIG DATA, RELATIONAL AND NON-RELATIONAL MODELS

В кінці минулого століття наука та технології вступили в нову еру – еру великих за обсягом даних. Науковий прогрес призвів до різкого збільшення кількості даних, що продукуються щодня. Наукові дані, зібрані на даний час, здебільшого залежать від ряду різних змінних. Тому їх можна представити у вигляді багатовимірних масивів або гіперкубів даних [1]. Впорядкування даних усередині гіперкубів є важливою задачею, і для цього було розроблено багатий спектр різноманітних інструментів.

Більшість сучасних програмних архітектур забезпечують підтримку обробки структур даних на базі гіперкубів [2]. Однак у кожній з цих програм гіперкуби даних можуть отримувати доступ до даних лише «ортогонально» до своїх «осей». Це означає, що можна вибирати лише певні значення або діапазони вздовж заданих розмірів. Це обмежує можливості аналізу даних, оскільки часто потрібно отримувати доступ до даних у неортогональних напрямках. Наприклад, може бути потрібно порівняти дані з різних осей або визначити, як зміна однієї змінної впливає на інші.

Поширення великих за обсягом даних створило потребу в масштабованих системах, які можуть ефективно їх обробляти. Реляційні бази даних, які базуються на SQL, є стандартним підходом до зберігання та керування даними. Вони добре працюють для структурованих даних, зокрема, таблиць, але мають обмеження щодо масштабованості. Бази даних NoSQL, з іншого боку, не використовують реляційну модель даних. Вони зазвичай краще підходять для керування великими обсягами даних, оскільки можуть масштабуватися по горизонталі, тобто формувати додаткові вузли для обробки даних.

Бази даних NoSQL – це нереляційні бази даних, які використовують різні моделі зберігання даних, що краще підходять для сучасних типів даних. На даний час існує чотири основні типи баз даних NoSQL: Бази даних типу «ключ-значення» зберігають дані у вигляді пари ключ-значення. Ключ ідентифікує запис, а значення містить дані про запис. Ці бази даних добре підходять для зберігання великих обсягів даних, які не мають чіткої структури. Бази даних документів зберігають дані у вигляді документів, які є JSON-подібними об'єктами. Ці бази даних добре підходять для зберігання неструктурованих даних, таких як текст і код. Бази даних стовпців зберігають дані у вигляді стовпців. Ці бази даних добре підходять для зберігання даних, які часто запитуються за певними стовпцями. Графові бази даних зберігають дані у вигляді графів. Граф складається з вузлів та ребер, які подають об'єкти та відношення між ними. Ці бази даних добре підходять для зберігання даних, що мають багато зв'язків між сутностями.

Захист особистої інформації користувачів є важливою задачею при передачі хмарних сховищ. Це пов'язано з тим, що хмарне сховище містить чутливі дані, зокрема, імена, адреси електронної пошти, номери кредитних карток та інші особисті дані.

Література

1. J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh, "Data cube: A relational aggregation operator generalizing group-by, cross-tab, and subtotals," Data mining and knowledge discovery, vol. 1, pp. 29–53, 1997.

2. Xtensor Stack, "Xtensor Documentation," In *Xtensor* (Version 0.24.6). Retrieved from Read the Docs: [https://xtensor. readthedocs.io/en/latest/], 2023.

УДК 004.9

Лебідко Д.М., Онуферко В.А., Перетятко Т.П.
Тернопільський національний технічний університет імені Івана Пулюя, Україна

ХМАРНІ ПЛАТФОРМИ, ОБЧИСЛЕННЯ ТА ІНТЕРНЕТ РЕЧЕЙ

Lebidko D.M., Onuferko V.A., Peretiatko T.P.

CLOUD PLATFORMS, COMPUTING AND THE INTERNET OF THINGS

Керування хмарною інфраструктурою та платформами є важливим та складним завданням, яке вимагає ретельного планування та поетапного виконання, а розуміння внутрішніх складнощів, які потрібно враховувати, допоможе установам та організаціям розробити ефективні плани управління. Існує багато внутрішніх складнощів, які потрібно враховувати.

Хмарні платформи повинні керувати хмарним програмним забезпеченням, яке використовується в установі чи організації. Це включає в себе такі завдання щодо встановлення, конфігурування та оновлення програмного забезпечення. Хмарна платформа повинна керувати транзакціями, передбаченими з постачальниками хмарних послуг. Це включає в себе завдання моніторингу відповідності транзакцій та вирішення спорів. Хмарна платформа повинна керувати хмарними програмними службами, які використовуються в організації. Це включає в себе завдання планування, розгортання та управління послугами. Хмарна платформа повинна сегментувати хмарну інфраструктуру на основі планових транзакцій. Це допомагає забезпечити, щоб дані, які підпадають під різні вимоги, були ізольовані одні від одних.

Хмарні обчислення – це модель обробки даних, при якій ресурси, сервери, сховища та програми, доступні через Інтернет. Це надає користувачам можливість отримувати доступ до цих ресурсів за потреби, не маючи необхідності інвестувати в власну інфраструктуру. Хмарні обчислення мають ряд переваг перед традиційними моделями обробки даних. Хмарні обчислення можуть бути більш економічно ефективними, ніж власна інфраструктура, оскільки користувачі сплачують лише за ті ресурси, які вони використовують. Хмарні обчислення можна легко масштабувати вгору або вниз відповідно до потреб, що може допомогти організаціям заощадити гроші. Хмарні обчислення доступні з будь-якого місця, де є Інтернет.

Інтернет речей – це мережа фізичних пристроїв, які підключені до Інтернету і можуть збирати та обмінюватися даними. Ці пристрої можуть використовуватися для широкого спектру цілей, таких як моніторинг стану обладнання, управління ланцюгами поставок та надання послуг клієнтам. Хмарні обчислення та IoT тісно пов'язані між собою. Хмарні обчислення можна використовувати для зберігання даних, які збирають пристрої IoT, а також для обробки цих даних [1]. Це дозволяє організаціям отримувати цінну інформацію з пристроїв IoT, яка може бути використана для покращення їхньої ефективності та продуктивності.

Хмарні обчислення дають користувачам змогу отримувати доступ до програм та даних через Інтернет. Це означає, що користувачі можуть використовувати свої «розумні» пристрої, смартфони та планшети, для доступу до своїх IT-ресурсів. Користувачі можуть отримувати доступ до своїх IT-ресурсів з будь-якого місця, де є Інтернет. Користувачі можуть отримувати доступ до своїх IT-ресурсів за допомогою одного інтерфейсу.

Література

1. Khadidos, A.O.; Shitharth, S.; Khadidos, A.O.; Sangeetha, K.; Alyoubi, K.H. Sangeetha, and Khaled H. Alyoubi. Healthcare Data Security Using IoT Sensors Based on Random Hashing Mechanism. J. Sens. 2022, 2022, 8457116.

Корба Д., Мудрик І.
ПРОЕКТУВАННЯ ТА РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ РУХОМИХ ОБ'ЄКТІВ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ JAVA, SPRING ТА ПРОТОКОЛУ GTFS
Korba D., Mudryk I.
DESIGN AND DEVELOPMENT OF MOBILE OBJECTS MONITORING SYSTEM USING JAVA, SPRING AND GTFS PROTOCOL TECHNOLOGIES 63

Віталій Кравчук
ПРОБЛЕМА ЗАХИСТУ КІБЕРПРОСТОРУ МАЛОГО ТА СЕРЕДНЬОГО БІЗНЕСУ
Vitaliy Kravchuk
CYBERSECURITY ISSUES FOR SMALL AND MEDIUM-SIZED BUSINESSES 64

О. Крамар, К. Козачук; Ю. Лавришук
КОНЦЕПТ VR-ПРОСТОРУ ЦЕНТРУ НАУКИ ТЕРНОПОЛЯ
O. Kramar, K. Kozachuk; Yu. Lavryshchuk
THE CONCEPT OF THE TERNOPIL SCIENCE CENTER'S VR SPACE 66

Т.О. Крамар, О.М. Дуда
ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ В «РОЗУМНОМУ МІСТІ»
T.O. Kramar, O.M. Duda
AUGMENTED REALITY TECHNOLOGIES IN THE SMART CITY 67

Крисюк М.В., Закопець А.І., Дуда Х.О.
СТАН ТА ПЕРСПЕКТИВИ ОБЧИСЛЮВАЛЬНИХ ПЛАТФОРМ ДЛЯ МІСЬКИХ ЗАДАЧ
Krysiuk M.V., Zakopets A.I., Duda Kh.O.
STATUS AND PROSPECTS OF COMPUTING PLATFORMS FOR URBAN TASKS 68

Крисюк М.В., Закопець А.І., Дуда Х.О.
ІНФОРМАЦІЙНО-ТЕХНОЛОГІЧНІ ПЛАТФОРМИ ТА ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ ДЛЯ ПОТРЕБ «РОЗУМНИХ МІСТ»
Krysiuk M.V., Zakopets A.I., Duda Kh.O.
INFORMATION TECHNOLOGY PLATFORMS AND SIMULATION FOR THE NEEDS OF SMART CITIES 69

Кубарич З.П., Скарга-Бандурова І.С.
ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ЕФЕКТИВНОГО РЕАГУВАННЯ НА ІНЦИДЕНТИ У SIEM СИСТЕМІ
Z.P. Kubarych, I.S. Skarga-Bandurova
USING ARTIFICIAL INTELLIGENCE FOR EFFECTIVE INCIDENT RESPONSE IN SIEM SYSTEM 70

О.П. Кузьмич, Я.В. Литвиненко
МЕТОДИ СТАТИСТИЧНОГО ОПРАЦЮВАННЯ МЕДИЧНИХ СИГНАЛІВ
O.P. Kuzmych, Ia.V. Lytvynenko
METHODS OF STATISTICAL PROCESSING OF MEDICAL SIGNALS 71

О.А. Кучеренко, О.О. Кучеренко
ОСОБЛИВОСТІ ПЕРЕДОБРОБКИ ДАНИХ ДЛЯ МЕТОДІВ ПРОГНОЗУВАННЯ
O.A.Kucherenko, O.O.Kucherenko
FEATURES DATA PREPROCESSING FOR FORECASTING METHODS 72

Лебідко Д.М., Онуферко В.А., Перетятко Т.П.
ВЕЛИКІ ЗА ОБСЯГОМ ДАНІ, РЕЛЯЦІЙНІ ТА НЕРЕЛЯЦІЙНІ МОДЕЛІ
Lebidko D.M., Onuferko V.A., Peretiatko T.P.
BIG DATA, RELATIONAL AND NON-RELATIONAL MODELS 73

Лебідко Д.М., Онуферко В.А., Перетятко Т.П.
ХМАРНІ ПЛАТФОРМИ, ОБЧИСЛЕННЯ ТА ІНТЕРНЕТ РЕЧЕЙ
Lebidko D.M., Onuferko V.A., Peretiatko T.P.
CLOUD PLATFORMS, COMPUTING AND THE INTERNET OF THINGS 74