УДК 004.65:519.1
Омельяненко І.О.[1]
[1] асп. НУ «Запорізька політехніка»

КОМПАРАТИВНИЙ АНАЛІЗ ІНСТРУМЕНТІВ ДЛЯ СТВОРЕННЯ ГРАФОВИХ БАЗ ДАНИХ

Сучасний розвиток технологій, таких як штучний інтелект та великі дані, вимагає нових підходів до зберігання та обробки даних. Графова база даних (ГБД) — це база даних, яка використовує структури графів для семантичних запитів з вершинами, ребрами та властивостями для представлення та зберігання даних. Реляційні бази даних часто не справляються з великим обсягом складних взаємозв'язків, що призводить до підвищеного інтересу до ГБД. ГБД не лише забезпечують ефективність у роботі з такими даними, але й пропонують високу масштабованість, що дозволяє обробляти складні структури даних у великих обсягах [1].

Серед популярних ГБД можна відзначити Neo4j, яка виділяється своєю продуктивністю та інтеграцією з різноманітними аналітичними інструментами [2]. GraphDB, зі свого боку, підкреслює важливість роботи з семантичними даними через RDF і підтримку запитів SPARQL [3]. Аналіз літературних джерел показує, що сучасні ГБД активно інтегруються із методами аналітики та онтологіями, що дозволяє виконувати складніші запити і забезпечувати більш точні результати [4].

Таблиця 1 – Характеристики найбільш популярних ГБД

База даних	**Підтримка моделей**	**Мова запитів**	**Сфери застосування**	**Масштабованість**
Neo4j	Property Graph	Cypher	Соціальні мережі, логістика, кібербезпека	Висока
GraphDB	RDF	SPARQL	Біоінформатика, управління знаннями	Висока
Datalevin	Datalog	Datalog	Локальні вбудовані системи	Обмежена
PachaDB	Нединамічні графи	Datalog	Розподілені системи, аналітика	Висока (залежить від конфігурації)

Результати дослідження показали, що Neo4j є найбільш придатною для обробки даних з великим обсягом складних взаємозв'язків [5]. Завдяки використанню Cypher як мови запитів, Neo4j забезпечує високий рівень продуктивності при обробці великих графів. Це робить Neo4j ідеальним вибором для проектів, пов'язаних із соціальними мережами, логістикою та кібербезпекою, де потрібно швидко аналізувати та обробляти великі масиви взаємопов’язаних даних [6].

GraphDB, що базується на RDF і підтримує SPARQL-запити, виявилася більш доречною для наукових досліджень, де потрібна точна семантична обробка даних. Її можливості в області семантики забезпечують більш інформативні результати, що є особливо важливим для онтологічних досліджень та проектів у галузі біомедицини [7].

У випадку з PachaDB, дослідження показало, що її підтримка незмінних даних є корисною для зберігання історії змін і забезпечення аудиту даних. Це особливо актуально для розподілених систем, де важливо зберегти цілісність та історію змін кожного запису [8].

Datalevin показала себе добре для локальних проектів, де обмеження масштабованості не є проблемою, але вимоги до гнучкості Datalog-запитів є високими.

Таблиця 2 – Переваги та недоліки існуючих ГБД

Інструмент	**Переваги**	**Недоліки**
Neo4j	Швидкість, гнучкість, потужні інструменти	Висока вартість ліцензії для корпоративних рішень
GraphDB	Потужна підтримка семантичних даних	Складність навчання SPARQL
Datalevin	Простота інтеграції, мінімальні ресурси	Обмежена масштабованість, проблеми з хмарною інтеграцією
PachaDB	Незмінність даних, висока надійність	Складність управління незмінними даними
Neo4j	Швидкість, гнучкість, потужні інструменти	Висока вартість ліцензії для корпоративних рішень

Проведене порівняння дозволяє зробити висновки щодо ефективності різних ГБД у конкретних сценаріях використання. Neo4j показала себе як ефективне рішення для корпоративних систем, що потребують високої швидкості та масштабованості [5]. Проте її недоліком є обмежена підтримка

семантичних запитів, що знижує її ефективність у проектах, де потрібно забезпечити семантичний аналіз даних.

GraphDB виявилася більш ефективною для наукових досліджень завдяки підтримці онтологій та можливості виконання семантичних запитів. Це надає цій базі перевагу в дослідницьких проектах, де важливо не тільки зберігати, але й інтерпретувати знання [3]. Проте її масштабованість може бути обмежена у порівнянні з Neo4j, що робить її менш ефективною у застосуваннях, де важлива швидкість обробки великих обсягів даних [1].

PachaDB та Datalevin демонструють свої переваги у локальних та розподілених системах зберігання. Однак, через обмежені можливості масштабування, їх застосування є доцільним у проектах, що не потребують швидкої обробки великих даних [4].

У підсумку, вибір ГБД залежить від специфічних вимог проекту. Neo4j підходить для високопродуктивних систем з великою кількістю взаємозв'язаних даних, GraphDB - для семантичного аналізу, а PachaDB та Datalevin можуть бути ефективними для збереження історичних даних [6].

Таблиця 3 – Критерії вибору інструменту для роботи з ГБД

Критерій	Neo4j	GraphDB	PachaDB
Продуктивність	Висока	Висока	Висока для розподілених систем
Простота інтеграції	Висока	Висока	Вимагає додаткових налаштувань
Підтримка складних запитів	Гнучкі запити через Cypher	Потужна робота з семантичними даними	Оптимізована для аналітики

Neo4j виявилася оптимальним вибором для систем, де важлива швидкість обробки запитів та гнучкість. Проте варто зазначити, що для роботи з онтологіями та семантичними запитами **GraphDB** є більш придатною. У подальших дослідженнях планується аналіз використання ГБД у хмарних системах та їх інтеграція з онтологічними структурами для оптимізації запитів і підвищення масштабованості [2].

Стратегія подальшого дослідження представлена на рисунку 1, який відображає ключові аспекти теми «Методи управління ГБД». Він містить такі основні розділи.

Типи графових баз даних: Property Graph, RDF і GraphDB, графи на основі Datalog. Це дає можливість дослідити різні типи моделей, які застосовуються для представлення ГБД.

Наявні методи управління: порівняння методів, практичні приклади (включно з соціальними мережами та біомедицинськими дослідженнями), оптимізація запитів для ефективнішої роботи з даними.

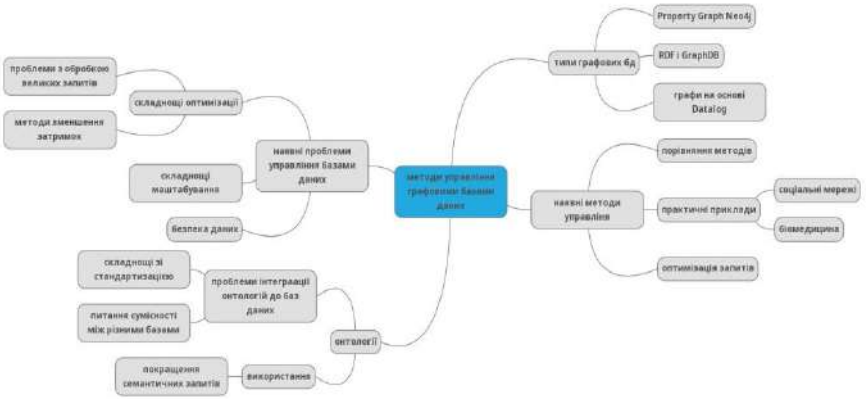

Рисунок 1 – Стратегія подальшого дослідження

Наявні проблеми управління: такі як складнощі оптимізації (обробка великих запитів, зменшення затримок), складнощі масштабування та забезпечення безпеки даних.

Проблеми інтеграції онтологій: стандартизація, сумісність між різними базами, а також покращення семантичних запитів для більш гнучкого використання онтологій у ГБД.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. https://cambridge-intelligence.com/choosing-graph-database/
2. https://linkurious.com/graph-database/
3. https://cgspace.cgiar.org/server/api/core/bitstreams/57ff8585-6e10-4fba-b220-e5779e33760c/content
4. https://dgraph.io/blog/post/data-ontology/
5. https://connect-lokesh.medium.com/ontology-driven-kg-construction-part-1-c29f9224dfdc
6. https://ceur-ws.org/Vol-3478/paper49.pdf
7. http://ujprm.com/index.php/rehabilitation/article/download/9/8
8. Blankenberg C., et al. Procedia Computer Science, 2022. https://doi.org/10.1016/j.procs.2021.12.019