УДК 004.6

Кравчук Р. Ю., студент 1 курсу спеціальності 122 «Комп'ютерні науки»
Науковий керівник:
Гончар В.М., асистент кафедри інформаційних технологій

ГРАФОВІ БАЗИ ДАНИХ. ЇХ РІЗНОВИД ТА ЗАСТОСУВАННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

Графові бази даних – це тип баз даних, що використовують графи для структурування і зберігання даних. У базах даних такого типу дані представлені, як вузли (вершини) з'єднані ребрами (відношеннями) , що дозволяє легко моделювати та зчитувати складні зв'язки. Ребро завжди має початковий вузол, кінцевий вузол, тип і напрямок. Крім того ребро може описувати «parent-child relationship», дії, право власності тощо. Немає обмежень щодо кількості та типу зав'язків, які може мати вузол. В базах даних такого типу відносини (ієрархія) між даними грає ключову роль. На *рисунку 1* показано приклад соціальної мережі. Враховуючи вузли (людей) та ребра (стосунки між ними), можна чітко зрозуміти, які в них стосунки та знайти, наприклад друзів-друзів конкретної людини [1].

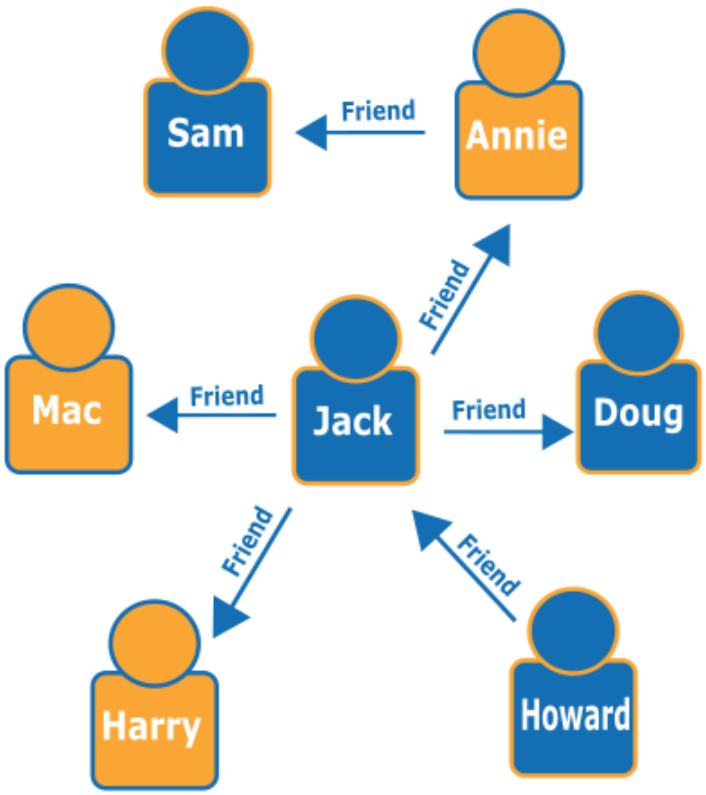

Рис. 1 Алгоритм роботи графової база даних

Графові бази даних не є лідерами по швидкості серед всіх видів баз даних, тим не менш сьогодні вони широко використовуються для вирішення наступних задач:

1. Низька затримка з великомасштабними даними. Коли ви додаєте багато зав'язків у реляційну базу даних, набори цих даних будуть величезними, а коли ви надсилаєте запит, його складність буде вищою і займе більше часу ніж зазвичай. Однак, графова база даних, була створена таким чином, щоб легко та швидко виконувати подібні завдання.
2. Коли зв'язки між елементами даних є важливіші. Наприклад є профілі, що містять різну інформацію, головною перевагою графових баз даних буде зв'язок між цими профілями, тобто, спосіб підключення до мережі.

Є 3 основні види графових баз даних: «Property graph», «Hypergraph» та «Triple store» кожен з, яких має своє призначення та способи використання[2].

«Property graph» (приклад Neo4j, AWS Neptune), дані в такому графі організовані, як вузли, зв'язки та властивості. Вузли — це сутності в графі. Вони можуть містити будь-яку кількість атрибутів (пар ключ-значення), які називаються властивостями. Вузли можна позначати мітками, що представляють їхні різні ролі у вашому домені. Мітки вузлів також можуть служити для додавання метаданих (таких як індекс або інформація про обмеження) до певних вузлів. Відносини забезпечують спрямовані, іменовані, семантично релевантні зв'язки між двома сутностями вузла. Зв'язок завжди має напрямок, тип, початковий вузол і кінцевий вузол. Як і вузли, відносини також можуть мати властивості. Завдяки ефективному способу зберігання зав'язків два вузли можуть спільно використовувати будь-яку кількість або тип зав'язків без шкоди для продуктивності. Хоча вони зберігаються в певному напрямку, зв'язки завжди можна ефективно переміщати в будь-якому напрямку.

«Hypergraph» (приклад HyperGraphDB) - це модель даних графа, у якій зв'язок («hyperedge») може з'єднувати може з'єднувати будь-яку кількість заданих вузлів. Це дозволяє будь-яку кількість вузлів на обох кінцях зв'язку. Це корисно, коли ваші дані містять велику кількість зав'язків.

У наведеному нижче прикладі *рисунок 4* ми бачимо, що Аліса та Боб є власниками трьох транспортних засобів, але ми можемо виразити цей зв'язок за допомогою одного «hyperedge». У графі властивостей ми повинні використовувати шість зав'язків, щоб виразити концепцію.

Оскільки «hyperedge» є багатовимірними, моделі гіперграфів більш узагальнені, ніж графи властивостей. Тим не менш, вони ізоморфні, тому ви завжди можете представити гіперграф як граф властивостей (хоча і з більшою кількістю зв'язків і вузлів), тоді як ви не можете зробити навпаки.

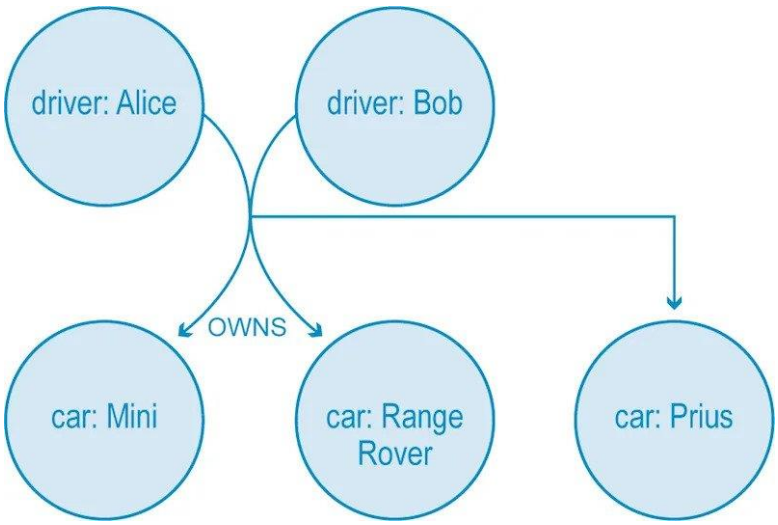

Рис. 4 Приклад моделі даних «Hypergraph»

«Triple Store» (наприклад, AWS Neptune, AllegroGraph) — потрійне сховище АБО RDF (Resource Description Framework) зберігає дані у форматі, відомому як потрійна структура даних суб'єкт-предикат-об'єкт. наприклад «Бобу 35» або «Боб знає Фреда». Додавання інформації представлено окремим вузлом. Зокрема, графова модель RDF складається з вузлів і дуг. Нотація графа RDF представлена — вузлом для суб'єкта, вузлом для об'єкта та дугою для предиката. Дані, оброблені потрійними сховищами, як правило, логічно пов'язані, тому потрійні сховища входять до категорії графових баз даних. Однак потрійні сховища не є рідними базами даних графівіц, оскільки вони не підтримують без індексну суміжність, а також їхні механізми зберігання не оптимізовані для зберігання графів властивостей. Потрійні сховища зберігають потрійні як незалежні елементи, що дозволяє їм масштабуватись горизонтально, але запобігає швидкому обходу зав'язків. Для виконання графових запитів потрійні сховища повинні створювати зв'язки з окремих незалежних фактів, додаючи затримку кожному запиту.

Через ці компроміси в масштабі та затримці найпоширенішим варіантом використання потрійних магазинів є офлайн-аналітика, а не онлайн-активність [3].

Проаналізувавши всю інформацію, що наведена вище, можна зробити висновок, що графові бази даних – це потужний інструмент, який сьогодні широко використовується для зберігання та оперування певними типами даних, що передбачають використання велику кількість зав'язків між файлами.

Список літератури

1. *Загальні поняття про графові бази даних URL:* *https://aws.amazon.com/nosql/graph/*
2. *Переваги графових баз даних URL:* https://www.geeksforgeeks.org/introduction-to-graph-databases/
3. *Види графових баз даних URL:* *https://neo4j.com/blog/other-graph-database-technologies/*