

УДК 004.9

## ПРОГРАМНІ ЗАСОБИ ОБРОБКИ ІЄРАРХІЧНИХ ДАНИХ

Б.Л. Шрамченко, кандидат технічних наук, с.н.с.

*Київський національний університет технологій та дизайну*

В.В. Ахматов, студент

*Київський національний університет технологій та дизайну*

Ключові слова: реляційна база даних, ієрархічні дані, скінчене поле, деревовидні структури, цілісність даних, рекурсивний запит, модель вкладених множин, документо-орієнтовані системи керування базами даних.

При створенні реляційних баз даних не передбачалися зручні засоби роботи з ієрархічними даними. Тому SQL бази не пропонують стандартного способу роботи з цими даними. Для роботи з графами, що відображають структуру ієрархічних даних, існують спеціальні бази даних. Наприклад Gremlin, SPARQL, Cypher. Але якщо в проекті немає можливості їх використовувати і потрібно працювати з ієрархічними структурами через SQL, то застосовують певні способи, які дозволяють вирішити цю проблему.

Мета поданої роботи - знайти удосконалений метод роботи з ієрархічними структурами, який буде водночас швидким та інтуїтивно зрозумілим. При цьому розв'язуються наступні задачі:

- 1) дослідити та порівняти різні методи та алгоритми, визначити швидкодію цих методів та вплив глибини ієрархічної структури на швидкодію.
- 2) розробити додаток для тестування методів роботи з різними деревовидними структурами в SQL.

Об'єктом дослідження у поданій роботі є особливості методів та алгоритмів, які дозволяють вирішувати різні проблеми роботи з ієрархічними даними в SQL. Предметом дослідження є визначення позитивних та негативних властивостей кожного методу, вибір найкращого та тестування швидкодії кожного методу.

Для визначення оптимального методу порівнюються такі критерії: швидкість роботи, складність додавання нового вузла у ієрархічну структуру, переміщення вузла, підтримка цілісності даних, видалення вузла. Робота з ієрархічними даними в реляційних базах даних виконується в багатьох застосунках, тому знаходження найкращого методу вирішить велику частину проблем, з якими зустрічаються розробники програмного забезпечення.

Найпопулярнішим методом для зберігання ієрархічних структур є метод Adjacency list [2]. Цей метод дозволяє зручно та наочно зберігати дані, не має обмежень за висотою дерева, що представляє ієрархічну структуру, дозволяє оптимізовано вставляти, видаляти та змінювати позицію вершини дерева, але призводить до проблеми, що висловлюється

у необхідності використання рекурсивних запитів, які підтримуються далеко не всіма СУБД. Для роботи з такими даними краще за все використовувати СУБД з підтримкою рекурсивних запитів, наприклад PostgreSQL. Якщо використовується база даних без рекурсивних запитів, як MySQL, необхідно використовувати тимчасові таблиці та процедури.

Метод Nested sets використовує модель вкладених множин і являє собою техніку для представлення дерев в реляційних базах даних [3]. Ідея цього методу полягає в зберіганні маршруту обходу дерева за префіксним порядком. При цьому спочатку обходять корінь дерева, потім, вузли лівого піддерева за префіксним порядком, і потім, вузли правого піддерева. Порядок обходу зберігають в полях left\_key та right\_key. У поле left\_key записують порядковий номер при вході в піддерево, а в right\_key - порядковий номер при виході з піддерева. Завдяки цьому можна обійтись без рекурсії.

Метод вкладених множин працює набагато швидше попереднього і дозволяє без рекурсії робити вибірки. Але він має суттєвий недолік, який полягає у необхідності додаткового обходу всього дерева при додаванні нової вершини або при переміщенні існуючої. Наприклад, при додаванні нової вершини в нижній рівень дерева, необхідно оновлювати всі поля left\_key та right\_key у всіх вершинах, які знаходяться правіше та вище тієї, що додається.

Метод Closure table передбачає використання дерева, що представляє собою вкладені підмножини, кореневий вузол включає в себе всі підмножини першого рівня, які в свою чергу включають в себе підмножини другого рівня і так далі. Для зберігання даних таким способом потрібні дві таблиці. В першу таблицю треба записати всі підмножини, в другу - список входжень кожної підмножини у батьківську та рівні підмножин.

Документо орієнтовані системи керування базами даних дозволяють за допомогою стандартної функціональності зручно та оптимізовано зберігати ієрархічні структури даних, але розробникам часто доводиться працювати з реляційними базами даних і не завжди є можливість використовувати документо орієнтовані. Тому проаналізовані методи та алгоритми для роботи з такими даними в реляційних базах. Кожний метод має свої переваги та недоліки в залежності від ситуації застосування.

#### Список використаних джерел

1. Celko J. Joe Celko's - Trees and Hierarchies in SQL for Smartie 2012. 2nd Edition. — Morgan Kaufmann, 2012. — 278 p.
2. Baka V. Python Data Structures and Algorithms. /B. Baka. Pact Publishing, 2017. - 310 p.
3. Голуб В.М. Ієрархічна модель вкладених множин у реляційних базах даних. ВІСНИК ЛЬВІВ. УН-ТУ. Серія прикл. матем. інформ. 2010. Вип. 16, с. 106-113.