

ВИКОРИСТАННЯ МОДЕЛІ NESTED SET ДЛЯ ЗБЕРІГАННЯ ДЕРЕВОВИДНИХ СТРУКТУР В БАЗІ ДАНИХ SQL

Перед автором була поставлена задача – зберігання та використання деревовидних структур з застосуванням технології MySQL при розробці веб-сайту оголошень besplatka.ua.

Стандартна реляційна алгебра та побудовані на ній SQL операції не можуть бути застосовані для всіх потрібних маніпуляцій з деревами (ієрархіями). Якщо дерево має довільну глибину, це не дозволяє використовувати SQL вирази для таких операцій, як порівняння місця в ієрархії для двох елементів або визначення належності елемента до певного під-дерева.

Існує декілька підходів для вирішення проблеми і деякі є доступними в системах керування базами даних:

- підтримка ієрархічних типів даних;
- розширення SQL для маніпуляцій з деревами;

SQL запити можуть бути виражені мовою програмування, яка підтримує ітерації та дозволяє виконувати реляційні операції.

Деревовидні структури в реляційних базах даних можна зберігати трьома основними способами: Adjacency List (суміжні списки), Materialized Path (матеріалізовані шляхи) та Nested Set (вкладені множини). Перший спосіб передбачає зберігання батьківського елемента в полі parent_id. Другий – повний шлях до елемента зберігається в полі типу path. Модель Nested Set передбачає зберігання дерева в трьох колонках – left, right та depth, де перші дві зберігають в собі діапазон усіх вкладених елементів.

Перші два способи незручні тим, що перехід по дереву з поточного елемента можна здійснити лише рівнем вище або нижче, або на кожен наступний рівень додаватиметься SQL запит, а тому це унеможливує перехід по дереву на довільну, заздалегідь невідому кількість рівнів. Хоча другий передбачає зберігання повного шляху елемента до кореня (тобто «хід» вверх), залишається та сама проблема з елементами нащадками.

А на відміну від вищеперерахованих методів, модель вкладених множин Nested Set дозволяє виконувати такі запити до бази даних. Ідея моделі полягає в нумерації вузлів відповідно до обходу дерева. Кожен вузол оброблюється двічі, кожному вузлу надається номер, відповідний до порядкового номеру згідно з обходом. Кожен вузол набуває двох номерів, які зберігаються як два атрибути.

Дерево складає собою масив вузлів, де кожен вузол має 4 складові – унікальний ідентифікатор, рівень вузла, лівий та правий ключ. Саме у цих двох цифрах закладена вся інформація про дерево. Щоб проставити ліві та праві ключі, краще всього накласти на дерево асоціацію з лабіринтом без циклів, який потрібно пройти від входу до виходу. Ідея полягає в тому, що потрібно йти лабіринтом, постійно та невідривно торкаючись стіни правою рукою. Таким чином здійснюється обхід від початку до кінця по всім вузлам лабіринту/дерева, причому захід в кожний елемент (комірка для правого та лівого ключів) здійснюється лише один раз. Головною особливістю такого підходу є те, що значення лівого (та правого, але для вибірки достатньо лівого) ключа будь-якого вузла (окрім кореня) знаходиться в діапазоні лівого та правого ключів усіх своїх предків, звідки власне пішла назва цієї моделі – «Вкладені множини».

Маючи на руках деревовидну структуру такого складу, можна здійснювати такі операції:

1. Побудова дерева – для цього потрібно лише відсортувати елементи за лівим ключем, тоді усі «діти» стануть за своїми «батьками», причому «сусіди» теж вистрояться в правильному порядку.
2. Вибір підпорядкованої гілки (усіх нащадків) – це ті елементи, ліві ключі яких знаходяться між лівим та правим ключами поточного елемента.
3. Вибір батьківської гілки (тобто усіх предків) – вибірка схожа на попередню, тільки навпаки – якщо лівий ключ поточного елемента знаходиться між лівим та правим певного вузла, то цей вузол – нащадок поточного.
4. Вибір гілки, в якій знаходиться поточний елемент – правий ключ таких елементів буде більше за лівий поточного, а лівий – менший за правий поточного.

У такої моделі є і недолік – кожна зміна в дереві – додавання, видалення або переміщення елемента веде за собою перебудову ключів усього дерева. Проте, кількість запитів до бази даних на зміну дерева не залежить від розмірів дерева та складає два – для додавання та видалення вузла, та три – для переміщення всередині дерева. Тому для дерев, які нечасто змінюються, але часто використовуються така модель, на думку автора, є найбільш оптимальною.