

УДК 004.658

DOI: 10.31673/2412-9070.2022.063237

Я. І. КОРНАГА, доктор техн. наук, професор;

А. О. БАРАБАШ, аспірант,

НТУУ «КПІ імені Ігоря Сікорського», Київ

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ ПІДВИЩЕННЯ ШВИДКОСТІ ПОШУКУ ТА ЗАПИСУ ІНФОРМАЦІЇ В БАЗАХ ДАНИХ

Розглянуто методи підвищення швидкості запису та пошуку інформації, а саме метод об'єднаних індексів та метод тимчасових таблиць. Основою першого методу є спеціальний індекс, який поєднує використання полів з однаковим типом та значенням із різних таблиць, а основою другого методу є використання тимчасової таблиці, в яку записуються дані та в якій відсутні індекси. Після закінчення роботи з цією таблицею вони переписуються в основну, а там уже відбувається перебудова індексів.

Розроблено опис алгоритмів, які розкривають роботу цих методів, та проведено аналітичну оцінку, яке дає змогу підтвердити ефективність застосування методів.

Для проведення експериментів розроблено спеціальний програмний комплекс для дослідження ефективності пошуку та запису інформації в БД, який можна адаптувати під різні операційні системи. Його застосування дало можливість повною мірою виконати експерименти та підтвердити аналітичні дані експериментальними.

Ключові слова: інформація; запис даних; пошук даних; індекси; бази даних; В+ -дерева; система керування базою даних.

ВСТУП

У сучасних системах керування базами даних (СКБД) процесами в роботі з інформацією є запис та пошук її в них. Основною характеристикою цього процесу є швидкість. У наявних СКБД під час кожного зберігання даних у таблицю відбувається перебудова індексів, що значно впливає на швидкість запису в БД. Користувачі мають витрачати час не тільки на закінчення попереднього збереження, а і на перебудову індексів. Відповідно до цього час затримки перед наступною операцією запису (модифікації) даних доходить навіть до кількох хвилин, що абсолютно не допустимо під час роботи баз даних в яких кілька сотень, а то і тисяч, клієнтів [1]. А якщо їх мільйони, то збереження буде відбуватися протягом досить тривалого часу, а продуктивність роботи сервера БД відповідно буде дуже швидко падати [2].

Постановка задачі. Потрібно створити нові методи підвищення ефективності запису та пошуку інформації в базах даних, які б дали змогу пришвидшити дані процеси та ефективніше використовувати ресурси СКБД.

ОСНОВНА ЧАСТИНА

Методи підвищення швидкості пошуку та запису в БД

Для підвищення ефективності швидкості пошуку однакових даних у різних таблицях було запропоновано метод об'єднаних індексів, який ефективно застосовується в тому разі, коли в базі даних присутнє використання полів з однаковим типом та значенням у різних таблицях [3].

У листі дерева такого об'єданого індексу потрібно зберігати запис-посилання на однакові поля в різних таблицях у форматі, як для звичайних В+-дерев, але з тією відмінністю, що запис окрім посилання на розміщення даних у таблиці повинен містити ще і посилання на саму таблицю, як це зображено на рис. 1.

ключ1 запис1.i	ключ 2 запис 2.j	...	ключ k запис k.m
----------------	------------------	-----	------------------

Рис. 1. Структура листової сторінки дерева об'єданого індексу

Значення запису-посилання $1, 2, \dots, k$ показують конкретне посилання на дані, а значення запису-посилання i, j, \dots, m — номер таблиці, в якій містяться дані.

Це дає можливість істотно економити час пошуку даних у різних таблицях одночасно, оскільки пошук буде здійснюватися по одному дереву.

Наведемо метод пошуку на основі алгоритму використання об'єднаних індексів.

1. Клієнт створює запит на пошук даних за кількома таблицями.

2. Запит обробляється компілятором та перевіряється на наявність об'єднаних індексів за даними полів та таблиць.

3. У разі наявності об'єднаного індексу, використовується він, а якщо об'єднаний індекс відсутній, то використовують стандартні табличні індекси для пошуку даних.

4. Клієнт отримує результати пошуку даних за кількома таблицями.

Для того, щоб не відбувалось затримок у роботі сервера БД, потрібно використовувати метод тимчасових таблиць для зберігання даних [4]. Потік інформації із запитами запису, який надходить від клієнтів, записуватиметься в них із тією умовою, що під час запису даних не відбувається перебудова індексів через їх відсутність у цій таблиці. Тимчасові таблиці створюватимуться для таблиць, в яких здійснюються операції запису даних найчастіше та в яких використовується багато індексів для проведення пошуку даних.

Тимчасові таблиці створюються на основі наявних основних таблиць та мають абсолютно ідентичний вигляд із ними. Єдиною відмінністю може бути додавання службових полів для визначення історії записів, в які можуть входити: час запису та ідентифікатор користувача, котрий проводив запис.

Перенесення даних із тимчасової таблиці в основну може відбуватися за чотирма варіантами:

- перенесення за кількістю полів заповнення тимчасової таблиці;
- перенесення за плином часу заповнення;
- перенесення в разі найменшої активності користувачів;
- перенесення на вимогу адміністратора (користувача) БД.

Для кращого розуміння роботи тимчасових таблиць опишемо їх життєвий цикл, унаочнює рис. 2.

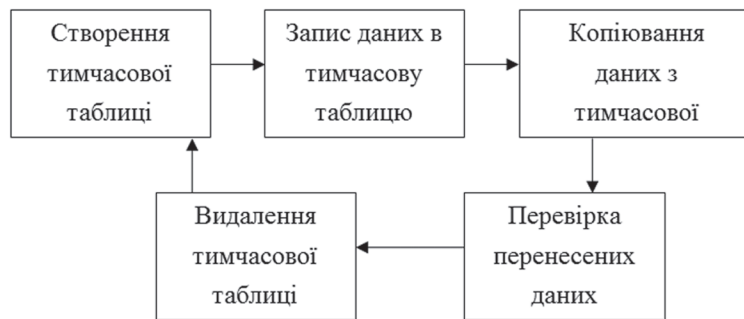


Рис. 2. Схема роботи тимчасових таблиць

Для того, щоб почати роботу з тимчасовими таблицями, СКБД створює їх у форматі, описаному раніше та фізично розміщує в робочому табличному файлі. Дані, які потрапляють на сервер від користувачів, записуються в них протягом часу доти, доки не настане момент копіювання даних в основні таблиці. Перед самим початком копіювання створюється нова тимчасова таблиця для того, щоб не переривалася робота з внесення даних користувачами СКБД. Після завершення копіювання даних в основну таблицю відбувається перевірка даних, що були перенесені. У разі, якщо все пройшло успішно, то відбувається перебудова індексів основної таблиці. Після проведення всіх раніше зазначених операцій тимчасова таблиця видалається через те, що на її місці вже працює нова тимчасова таблиця, в яку вже заповнюються нові дані.

З огляду на сказане опишемо метод запису (модифікації) даних на основі алгоритму використання методу тимчасових таблиць відповідно до варіантів перенесення даних до основних таблиць.

1. Клієнт створює запит на запис даних.
2. Компілятор обробляє запит та визначає таблиці, в які потрібно проводити запис (модифікацію).
3. Якщо для таблиці, в яку потрібно проводити запис, існує тимчасова таблиця, то запис проводиться в неї, а якщо відсутня, то запис здійснюється в основну таблицю.
4. Якщо дані, які потрібно модифікувати, містяться в основній таблиці, то виконується модифікація даних та перебудова індексів основної таблиці, а якщо дані, які потрібно модифікувати, не було знайдено в основній таблиці, то проводиться пошук і модифікація даних у додатковій таблиці.

Після завершення запису (модифікації) клієнту передається кількість даних, які були записані (модифіковані) [5].

Аналітичне оцінювання параметрів методів запису та пошуку даних

Під час розрахунку часу пошуку даних за методом об'єднаних індексів візьмемо до уваги той факт, що, як зображено на рис. 3, пошук може відбуватися за двома варіантами. Для першого варіанта характерно те, що пошук посилає на дані з різних таблиць у дереві відбувається по тих самих вузлах, а отже заощаджується час \dot{O}_A на зчитування з жорсткого диска вузлів індексу (дерева) для однієї з таблиць, завдяки тому, що їх уже було зчитано для другої таблиці. Відповідно час пошуку \dot{O}_{II} розраховуватиметься за формулою

$$\dot{O}_{II} = m(\dot{O}_A + 2\dot{O}_{\bar{A}}), \tag{1}$$

де \dot{O}_A — час зчитування вузлів дерева індексу пошуку з жорсткого диска; $\dot{O}_{\bar{A}}$ — час пошуку у вузлі дерева для першої та другої таблиць; m — висота дерева.

Для другого варіанта характерно те, що пошук посилянь на дані з різних таблиць у дереві відбувається не по тих самих вузлах, тож заощаджується час тільки на зчитування з жорсткого диска вузлів індексу (дерева) для однієї з таблиць до того моменту, поки шляхи пошуку по дереву не розійдуться. Відповідно час пошуку \dot{O}_I обчислюватиметься за виразом:

$$\dot{O}_I = m(\dot{O}_A + 2\dot{O}_{\bar{A}}) + d\dot{O}_{\bar{A}}, \tag{2}$$

де \dot{O}_A — час зчитування вузлів дерева індексу пошуку з жорсткого диска для першої таблиці; $\dot{O}_{\bar{A}}$ — час пошуку у вузлі дерева для першої та другої таблиць; m — висота дерева; d — кількість вузлів, що додатково потрібно зчитати з жорсткого диска ($d < m$).

Отже, для розрахунку часу пошуку даних за методом об'єднаних індексів потрібно використовувати формулу (2), а в разі відсутності розбіжності у вузлах пошуку $d = 0$ та час пошуку розраховується за першим варіантом.

Фактично відбувається економія часу пошуку даних, що становить: для першого варіанта розрахунку $m\dot{O}_{\bar{A}}$, а для другого варіанта — $(m - d)\dot{O}_{\bar{A}}$.

Для оцінювання параметрів запису інформації в БД за допомогою методу тимчасових таблиць потрібно ввести такі позначення:

- \dot{O}_C — час запису інформації в БД;
- \dot{O}_{ID} — час однієї перебудови індексів для відповідної таблиці;
- \dot{O}_{CO} — час одного запису інформації;
- \dot{O}_{IA} — час перенесення даних із тимчасової таблиці в основну;
- n — кількість записів, які проводяться у відповідну таблицю.

Взагалі, в процесі запису даних час запису визначається за рівністю

$$T_3 = n(T_{3T} + T_{3P}), \tag{3}$$

тобто, під час кожного запису витрачається час на сам запис інформації \dot{O}_{CO} та час на перебудову індексів \dot{O}_{ID} .

Для методу використання тимчасових таблиць характерною особливістю є відсутність багаторазової перебудови індексів для таблиці, тому формула часу запису інформації в БД матиме такий вигляд:

$$T_3 = nT_{3T} + T_{3P} + T_{3D}, \tag{4}$$

тобто, замість виконання n перебудов індексів ми перебудовуємо їх тільки одноразово.

Оскільки час \dot{O}_{ID} перебудови індексів значно більший за час \dot{O}_{IA} запису з тимчасової таблиці в основну, то \dot{O}_{IA} можна знехтувати, а виграш часу запису даних із використанням методу тимчасових таблиць становитиме $(n - 1)\dot{O}_{ID}$, що дає можливість виконувати більшу кількість операцій запису одночасно.

Експериментальне оцінювання параметрів методів запису та пошуку даних

Для проведення експериментів було розроблено спеціальний програмний комплекс для дослідження ефективності пошуку та запису інформації в БД, який можна адаптувати під різні операційні системи. Для виконання експерименту з ефективності застосування індексів на основі методу об'єднаних індексів у базах даних було вибрано операційну систему Linux Ubuntu [1].

У процесі здійснення експерименту було проведено порівняння швидкості пошуку за індексами на основі В+-дерев та індексами побудованими на основі методу об'єднаних індексів (рис. 3).

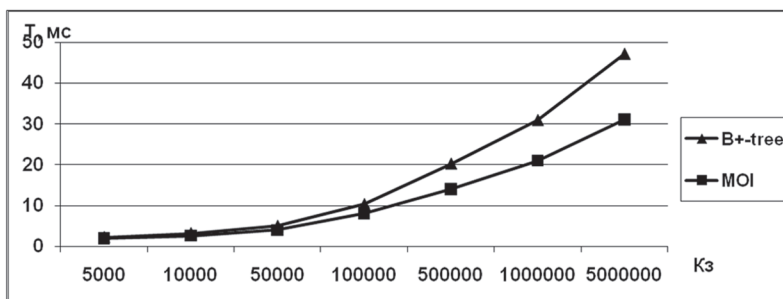


Рис. 3. Залежність часу пошуку від кількості записів для методів формування індексів у БД за допомогою В+-дерев та МОІ

Загалом ефективність використання методу об'єднаних індексів, як показано на рис. 3, для заданої ОС на заданому проміжку кількості записів у таблицях бази даних становить майже 25% порівняно з В+-деревами, що дає значне пришвидшення в роботі бази даних та дає змогу витрачати її ресурси на інші потреби.

Середньоквадратичне відхилення становило для В+-дерев майже 0,91%, а для методу об'єднаних індексів — 0,92%, з чого можна дійти висновку, що експеримент було проведено в межах допустимих правил, а здобутий результат відповідає потрібній точності [6].

Для реалізації експерименту з ефективності застосування методу тимчасових таблиць в базі даних було створено тимчасову таблицю, за якою не будувалися індекси. У цю таблицю додавалися записи простим дописуванням їх в її кінець. А потім під час заповнення таблиці на відповідну кількість записів вона копіювалася в основну таблицю і вже після збереження скопійованих даних відбувалася перебудова індексу. Експеримент проводився на операційній системі Linux Ubuntu.

Отже, завдяки такому експерименту визначилися час запису в таблицю, час копіювання в таблицю, час перебудови індексу. Також, застосувавши формулу (5) розрахунку загального часу для проведення n операцій запису в БД, було здобуто графік (рис. 4) в програмі MS Excel, апроксимація якого і дала функцію, за допомогою якої можна розрахувати час побудови індексу для відповідної кількості записів в таблиці [1; 7].

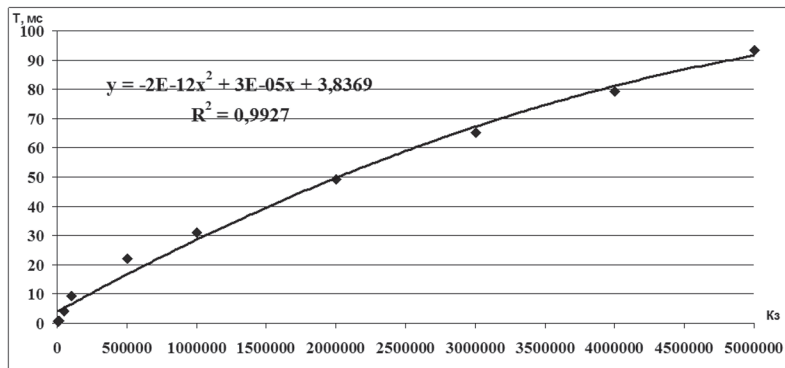


Рис. 4. Час побудови індексів залежно від кількості записів у таблиці

Відповідно формулу для визначення екстрапольярних даних можна подати у вигляді

$$y = -2E - 12x^2 + 3E - 0,5x + 3,8, \quad (5)$$

де x — це кількість записів; y — час, за який відбувається перебудова індексу.

А для визначення часу побудови індексів для відповідної ділянки кількості записів інтегруємо (5) та дістанемо

$$y = -2E - 12x^3/3 + 3E - 0,5x^2/2 + 3,8x. \quad (6)$$

Отже, маючи вираз (6), можна розрахувати (таблиця) сумарний час запису відповідної кількості записів в основну таблицю та в тимчасову таблицю.

Порівняння часу запису в БД із застосування методу тимчасових таблиць та без його застосування

Кількість записів	Час запису у файл без застосування методу тимчасових таблиць	Час запису у файл із застосуванням методу тимчасових таблиць	Логарифм часу запису без МТТ	Логарифм часу запису з МТТ
5000	5,15	4,85	0,69	0,71
10000	14,60	7,65	0,88	1,16
50000	64,97	31,35	1,50	1,81
100000	279,73	82,38	1,92	2,45
500000	1966,67	294,28	2,47	3,29
1000000	6733,33	681,80	2,83	3,83
2000000	23866,67	1229,82	3,09	4,38
3000000	49800,00	1765,91	3,25	4,70
4000000	82933,33	2300,10	3,36	4,92
5000000	121666,67	2843,60	3,45	5,09

Оскільки порядок даних значно відрізняється для різної кількості записів, то для кращого унаочнення побудуємо графік залежності кількості записів від сумарного часу запису в таблицю, узявши десятковий логарифм від сумарного часу (рис. 5).

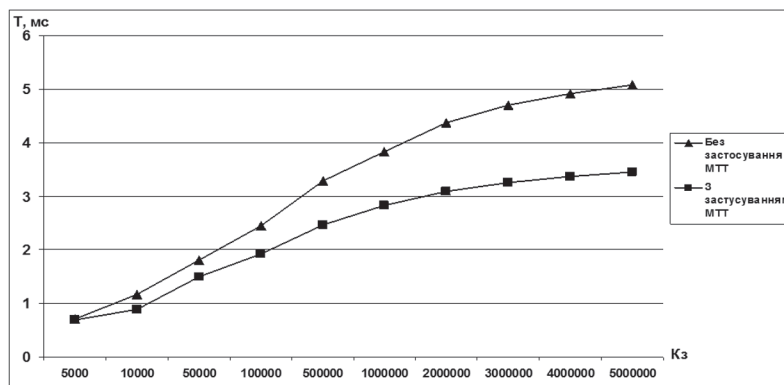


Рис. 5. Логарифмічна функція часу додавання відповідної кількості записів у таблицю без застосування та із застосуванням МТТ

Аналіз рис. 5 свідчить про ефективність застосування тимчасових таблиць у базах даних, які не містять індексів.

ВИСНОВКИ

Пошук на основі методу об'єднаних індексів доцільно використовувати для тих таблиць, які найчастіше підлягають пошуку та поля в яких найчастіше порівнюють між собою. Зазвичай пошук буде використовуватися для двох таблиць, але зі збільшенням обсягів бази даних та розподіленістю її елементів він може використовуватися і для більшої кількості таблиць. Використання методу хешування індексів зумовлює покращення часу пошуку та швидкості перебудови індексів під час запису (модифікації) даних у базу даних у тих випадках, коли той самий індекс застосовується в багато разів частіше за короткий проміжок часу на 25%. Водночас використання методу тимчасових таблиць для запису даних у БД спричинює значне зменшення сумарного часу запису через відсутність у тимчасовій таблиці БД індексів та потребу в їх перебудові.

Отже, вибір певного методу запису та пошуку даних залежить від кількох параметрів побудови бази даних, а саме: кількості таблиць та записів у них, кількості звернень до відповідної таблиці, кількості побудованих індексів на відповідну таблицю та типів даних у відповідних полях бази даних.

Список використаної літератури

1. Корнага Я. І. Порівняльна оцінки застосування методів підвищення швидкості пошуку та запису даних в базах даних // *Адаптивні системи автоматичного керування*. 2013. № 1. С. 37–44.
2. Hirt M., Lagergren M. *Oracle jrookit: The Definitive Guide* // Packt Publishing, 2010. P. 588.
3. Білова Т. Г. Проектування розподіленої бази даних системи надання електронних адміністративних послуг // *АСУ та прилади автоматики*, 2019. № 176. С. 49–54.
4. Бальченко І. В. Проблеми розроблення неоднорідних розподілених систем керування базами даних // *Технічні науки та технології*, 2016. № 2 (4). С. 67–71.
5. Мухін В. Є., Корнага Я. І. Механізми підвищення ефективності процедури моніторингу безпеки в розподілених базах даних: зб. наук. праць // *Вісник Нац. техн. ун-ту «Харківський політехнічний інститут»*. 2012. № 38. С. 128–135. (Серія: Інформатика та моделювання).
6. Program Code Protecting Mechanism Based on Obfuscation Tools / V. Mukhin, V. Zavgorodnii, Y. Kornaga [et al.] // *IEEE International Conference on System Analysis & Intelligent Computing*. 2022. P. 407–419.
7. Method of Restoring Parameters of Information Objects in a Unified Information Space Based on Computer Networks / V. Mukhin, V. Zavgorodnii, O. Barabash [et al.] // *International Journal of Computer Network & Information Security*. 2020. 12 (2). P. 11–21.

Y. I. Kornaga, A. O. Barabash

COMPARATIVE ANALYSIS OF METHODS FOR INCREASING THE SPEED OF SEARCHING AND RECORDING INFORMATION IN DATABASES

The work examines the methods of increasing the speed of writing and searching information, namely the method of combining indexes and the method of temporary tables. It is noted that today it is necessary to create new methods that can increase the efficiency of records and search for information in databases, which would allow speeding up the mentioned processes and more efficiently using DBMS resources.

The basis of the first method in the work is a special index that combines the use of fields with the same type and value from different tables. The basis of the second method is the use of a temporary table in which data is written and in which there are no indexes. It is noted that the method of combining indexes is effectively used when the database uses fields with the same type and value in different tables. After finishing work on this table, they are rewritten in the main one, and the indexes are already being rearranged there.

A description of the algorithms describing the operation of these methods was developed, and an analytical evaluation was carried out to confirm the effectiveness of the methods.

For conducting experiments, a special software package was developed for researching the effectiveness of searching and recording information in the database, which can be adapted to different operating systems (Linux, Ubuntu, etc.). Its use made it possible to fully perform experiments and confirm analytical data with experimental ones.

It is highlighted that the search based on the method of combined indexes is advisable to use for those tables that are most often searched and the fields in which they are most often compared with each other. It is emphasized that the choice of a certain method of recording and searching for data depends on several parameters of the database construction, namely: the number of tables and records in them, the number of references to the corresponding table, the number of constructed indexes on the corresponding table, and the types of data in the corresponding fields of the database.

Keywords: information; data recording; data retrieval; indexes; databases; B+ -trees; database management system.

