

Список використаних джерел

1. Згуровский М. Определение университетских рейтингов – составляющая евроинтеграции в образовательной сфере // Зеркало недели – Украина, №28, 2006 – С. 10-13.
2. Шаров Ю.П. Возможности альтернативного подхода к оценке успешности деятельности вуза с помощью рейтингования // Ранжирование высших учебных заведений: состояние, тенденции и проблемы. Коллективная монография / Под редакцией В.Н.Бержанского. – Симферополь: ДИАЙПИ. – С.52 – 54.
3. Ямковий В. Ранжування університетів – крок до відкритості та прозорості вищої освіти // Освіта України – Україна, № 42, 5 червня 2007. – С.4-7.

УДК 004.657

ОПТИМІЗАЦІЯ ПОШУКУ ІНФОРМАЦІЇ В БАЗІ ДАНИХ НА ОСНОВІ ІНДЕКСІВ

Сигінь Ю.С.¹⁾, Яцків Н.Г.²⁾

Тернопільський національний економічний університет

^{1)магістр; ^{2)к.т.н., доцент}}

I. Постановка задачі

Продуктивність є одним з найважливіших показників, яка вказує на те чи проект є успішним, чи помилковим. Під продуктивністю певної системи розуміється час її реакції на певні дії користувача і повернення користувачеві результату роботи (для бази даних, це швидкість отримання вибірки даних). З цього можна зробити висновок, що найбільш доцільно проектувати систему і використовувати певні технології на самих ранніх етапах розробки продукту.

Сучасні СУБД обробляють велику кількість різноманітної інформації, яка зберігається в базі даних. Об'єми інформації можуть сягати від декількох рядків таблиці до декількох гігабайтів. При цьому виконання запитів до бази даних має бути оперативним, а час відповіді повинен бути мінімальним. Для забезпечення такого режиму роботи СУБД повинна використовувати певні технології для прискорення вибірки даних і повернення їх користувачу.

При пошуку рядків в таблиці за допомогою оператора SELECT, дана інформація послідовно завантажується в оперативну пам'ять сервера. Якщо база даних є невеликою, тоді час повернення результату запиту можна й не помітити, але при відносно великих об'ємах бази даних продуктивність значно падає, оскільки неможливо завантажити всю таблицю в оперативну пам'ять, що в свою чергу призводить до збільшенню дискових операцій вводу/виводу [1].

Одним із найбільш поширених шляхів вирішення проблеми високої продуктивності являється використання індексів. Індекс по своїй структурі пришвидшує виконання запиту до бази даних, тим самим надаючи користувачу швидкий доступ до потрібної інформації. Індекс утримує в собі записи, які складаються з ключового поля по якому був сформований індекс і відповідні адреси на рядки таблиць [3]. Індекс значно менший в розмірах за таблицю, що призводить до зменшення кількості дискових операцій і збільшення часу відклику.

Сучасні веб-ресурси повинні оперативно взаємодіяти з користувачами і робити це якнайшвидше. Наприклад, користувачі мобільних пристроїв для отримання розкладу потягів по певній станції, формують одночасно багато запитів до одного веб-сервера, а саме бази даних, яка має швидко дати результат роботи. Отже, оптимізація запитів і використання певних технологій для збільшення продуктивності бази даних, являється актуальною задачею.

II. Мета роботи

Метою роботи є аналіз ефективності пошуку інформації в базі даних на основі індексів для створення високопродуктивної бази даних для пошуку даних.

III. Кластеризований індекс

Кластеризований індекс - це деревоподібна структура даних, при якій значення індексу зберігається разом з даними, які відповідні індексу. І індекси, і дані при такій організації впорядковані. При додаванні нового рядка в таблицю, вона дописується не в кінець файлу, а в потрібну гілку деревовидної структури, відповідну їй по сортуванню. Слово файл при використанні кластеризованих індексів являє собою деяку одиницю виміру даних, яка відповідає одній таблиці.

Найбільш продуктивним движком для MySQL є InnoDB, оскільки він дозволяє використовувати кластеризовані індекси. Дані в InnoDB зберігаються по файлах (таблицях), які мають розмір 16 Кб. Розмір однієї таблиці – це розмір вузла деревоподібної структури, від якого залежить коли почнеться розгалуження [2]. Коли дані перестають поміщатися в один файл, тоді починається розгалуження таблиці і формуються дві нові таблиці з даними і індекс, який поєднує дані таблиці (рисунок 1).

Кластеризований індекс значно пришвидшує вибірку результатів з бази даних, його потрібно використовувати коли дані часто не змінюються, бажано використовувати індекс для первинного ключа, тобто поле має бути найбільш коротким, кластеризований індекс потрібно використовувати з відсортованими даними.

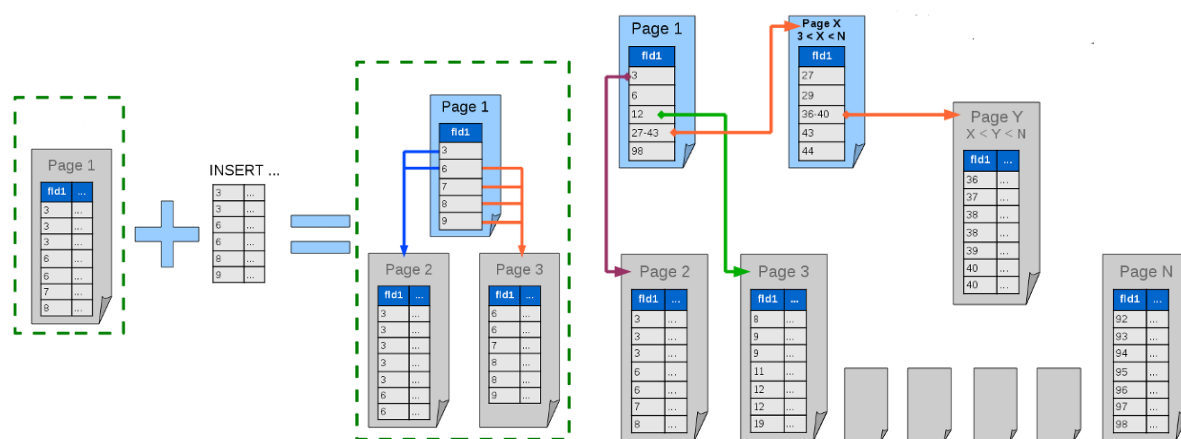


Рисунок 1 – Кластерне представлення таблиці

IV. Некластеризований індекс

На відміну від кластеризованого індексу, листя некластеризованого індексу містять тільки ті стовпці (ключові), за якими визначено даний індекс, а також містить вказівник на рядки з реальними даними в таблиці. Це означає, що системі необхідна додаткова операція для виявлення і отримання необхідних даних. Некластеризований індекс не може бути відсортований на відміну від кластеризованого, однак він дозволяє створити більш ніж один індекс в таблиці [4]. Але це не означає, що потрібно створювати безліч індексів, індекси можуть, як пришвидшити час виконання запитів, так і сповільнити. Некластеризований індекс слід використовувати для запитів, які повертають невелику кількість даних і стовпців. Для великих наборів даних слід використовувати кластеризовані індекси.

Висновок

Виходячи з вищенаведених технологій оптимізації пошуку інформації в базі даних маршрутів потягів, найбільш доцільно використати кластеризований індекс для збільшення продуктивності бази даних. Оскільки в даній базі даних номер маршруту і станції, використовується найчастіше для формування запитів, а дані поля являються унікальними. Для того, щоб інформація займала менше місця і була більш структурованою при передачі користувачу, необхідно дані конвертувати в JSON масив.

Список використаних джерел

1. Казакова И. А., Фролов К. М. Оптимизация доступа к данным на основе индексов // Мир современной науки. 2014. №3 (25).
2. CeciliaCioloca, Mihai Georgescu, M. Y. Increasing Database Performance using Indexes. Database Systems Journal vol. II, no. 2/2011
3. Гарсия-Молина, Г., Ульман, Д. Д., Уидом, Д. Системы баз данных : Полный курс [Текст] / Гектор Гарсия-Молина, Джефри Д. Ульман, Дженифер Уидом ; пер. с англ. – М. : Издательский дом «Вильямс», 2012.
4. 14 Вопросов об индексах в SQL Server, которые вы стеснялись задать [Электронный ресурс] – Режим доступа до ресурсу: <https://habrahabr.ru/post/247373/>.