JOURNAL
COMPUTER SCIENCES INSTITUTE

JCSI 28 (2023) 186-189
Received: 16 April 2023
Accepted: 29 June 2023

The Examination of SQL Queries Efficiency in Chosen IT System

Badanie wydajności zapytań SQL w wybranym systemie informatycznym

Krzysztof Barczak*

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract
Optimization of SQL queries is an important part of any system using a database. Valuable time can be gained by consciously making changes within the database. Therefore, several different optimization methods have been selected and resented in this article. The research was conducted on the MS SQL database engine. The execution times of SQL queries were carefully examined according to defined criteria, then performance optimization was done, followed by repeated tests to obtain results after the optimization. Finally, the results were compared through which conclusions were drawn.

Keywords: SQL query optimization; databases; testing; performance analysis

Streszczenie
Optymalizacja zapytań SQL jest istotnym elementem każdego systemu, wykorzystującego bazy danych. Świadomie dokonane zmiany w obrębie bazy danych pozwalają zyskać cenny czas. Dlatego też w niniejszym artykule wybrano i przedstawiono kilka różnych metod optymizacji, natomiast badania zostały przeprowadzone na silniku bazodanowym MS SQL. Czasy wykonania zapytań SQL dokładnie zmierzono według zdefiniowanych kryteriów, a następnie przeprowadzono optymalizację wydajnościową oraz powtórzono badania w celu ponownego określenia czasu wykonania zoptymalizowanych zapytań. Na podstawie porównanych wyników zostały wyciągnięte odpowiednie wnioski.

Słowa kluczowe: Optymalizacja zapytań; bazy danych; testowanie; analiza wydajności

*Corresponding author

Email address: krzysztof.barczak@pollub.edu.pl (K. Barczak)

1. Wstęp

Bazy danych występują wszędzie. W każdej firmie, która wykorzystuje dane muszą być one przechowywane w odpowiedni sposób. Zależnie od dostępności do systemów informatycznych, możliwości finansowych czy też chęci, pracownicy zauważają, w których obszarach program potrzebuje więcej czasu na odpowiedź z bazy danych, a gdzie wyniki zapytań są otrzymywane od razu. Jeżeli firmie zależy na prawidłowym korzystaniu z systemu powinna ona zwracać również uwagę na pojawiające się w nim problemy wydajnościowe.

Dzięki zmianom zapytań SQL można zaoszczędzić czas, co oczywiście przekłada się na zyski. Z doświadczenia autora wynika, że firmy, którym zależy na płynnym działaniu systemów informatycznych, osiągają lepsze wyniki. Dlatego tak ważne jest pojęcie optymalizacji zapytań SQL i utrzymania prawidłowego działania serwera. Należy uważać nawet na z pozoru rzadko występujące problemy funkcjonowania serwera bazy danych, takie jak brak miejsca na serwerze produkcyjnym. Skrócenie czasu zapytania zapobiega blokowaniu wykonania polecenia przez drugie (ang. deadlocks) oraz zwiększa wydajność działania aplikacji, co prowadzi do sytuacji „win-win", gdzie w tym samym czasie z aplikacji może korzystać więcej osób, bez konieczności rozbudowy serwera.

Wybranie odpowiednich kolumn podczas przygotowywania zapytań sprawia również, że czas wykonania zapytania skraca się. Ważnym aspektem są również indeksy. Umiejętne ich dobranie potrafi znacząco skrócić czas oczekiwania odpowiedzi na zapytanie. Należy mieć jednak na uwadze, że nadmiarowa liczba indeksowanych kolumn może negatywnie wpłynąć na pracę bazy danych. Konieczna jest więc analiza potrzeb informacyjnych i optymalne dobranie kolumn, które będą indeksowane.

Oczywiście równie istotnym czynnikiem są parametry serwera. Niewłaściwie dobrana konfiguracja sprzętowa i oprogramowanie potrafią znacząco obniżyć wydajność silnika bazodanowego jak i również aplikacji korzystającej z niego.

2. Przegląd literatury

Wybranie odpowiednich kolumn podczas definiowania zapytania sprawia, że czas jego wykonania skraca się. Dostępne są publikacje, w których dokonano analizy i przedstawiono zagadnienia odnoszące się do badań czasów wykonywania zapytań SQL.

Pierwszym przykładem jest artykuł naukowy „Analiza wydajności relacyjnych baz danych Oracle oraz MSSQL na podstawie aplikacji desktopowej" przygotowany przez Grzegorza Dziewita, Jakuba Korczyńskiego i Marię Skublewską-Paszkowską [1]. W artykule poruszona jest kwestia czasów prostych zapytań dla następujących rekordów: 10, 30, 50, 10000, 50000, 100000 dla danych tekstowych oraz numerycznych. Głównym celem badania było porównanie dwóch silników bazodanowych: MS SQL oraz Oracle. Wnioski okazały się następujące: baza Oracle szybciej wykonuje

operację SELECT, jednak baza MS SQL szybciej realizuje operacje UPDATE oraz INSERT.

Drugą, nieco starszą pozycją jest artykuł naukowy „Optymalizacja Microsoft SQL server przy współpracy z Microsoft Dynamic NAV" przygotowany przez Marcina Wocha i Piotra Łebkowskiego. Artykuł ten przedstawia narzędzia stworzone do wspierania administratorów systemu NAV. Narzędzia umożliwiają optymalizację kodu oraz monitorowanie systemu [2]. W tej publikacji potwierdzono iż prawidłowa administracja serwerem oraz systemem Navision pozwala na optymalizację zapytań oraz manipulacji danymi.

Kolejnym przykładem jest pozycja „Pro T-SQL 2019" [3]. Autor tej publikacji opisuje wiele aspektów pisania kodu T-SQL oraz jego optymalizacji. Jednym z nich jest optymalizacja logicznych odczytów (ang. Logical Reads). Jednym z wniosków jest poprawa wydajności wykonania kodu T-SQL, dzięki dobrym praktykom projektowym.

Warto również wspomnieć o badaniach przeprowadzonych na bazach nie SQL-owych [4]. Wybrana pozycja „Analiza prędkości wykonywania zapytań w wybranych bazach nie SQL-owych" przedstawia badania oraz analizę wyników czasów zapytań dla baz nierelacyjnych takich jak CouchDB i MongoDB. Głównym, generalnym wnioskiem jest stwierdzenie iż baza MongoDB jest wydajniejsza względem bazy CouchDB.

Powyższe pozycje literaturowe pokazują sposób badania czasów zapytań oraz wskazują na różne czynniki, które mogą zostać poddane optymalizacji w celu uzyskania krótszych czasów wykonania zapytań SQL. Za takie aspekty uważa się między innymi korzystanie z tabel tymczasowych, tworzenie indeksów czy wybieranie tylko potrzebnych danych. Podczas przeglądu literatury zauważono, iż brakuje badań przedstawiających optymalizację czasów wykonywania zapytań SQL dla bazy danych MS SQL Server.

3. Cel i zakres badań

Celem badań było przeprowadzenie analizy wpływu optymalizacji zapytań SQL na szybkość ich wykonywania oraz udowodnienie, iż różne metody optymalizacyjne potrafią skrócić czas wykonania zapytania. Z badań nad możliwościami optymalizacyjnymi zapytań SQL wyciągnięto odpowiednie wnioski, tak aby pomóc w doborze najbardziej optymalnych ustawień oraz konfiguracji serwera MS SQL. Przeprowadzona analiza wpływu optymalizacji zapytań SQL pozwoliła na postawienie następujących tez badawczych:

T1: Wybór tylko niezbędnych kolumn w zapytaniu SQL pozwala na zwiększenie wydajności zapytań.

T2: Poprzez usunięcie klauzuli „order by" nastąpi skrócenie czasu trwania zapytania SQL.

T3: Skrócenie czasu trwania zapytań SQL nastąpi poprzez dodanie odpowiednich indeksów.

T4: Rodzaj złączenia między tabelami wpływa na czas wykonania zapytań SQL.

4. Metodyka badawcza

Do przeprowadzenia badań wydajności zapytań SQL przygotowano aplikację na wzór znanego portalu zajmującego się sprzedażą samochodów [5]. Do wykonania aplikacji testowej wykorzystane zostały: szkielet programowania Symfony oraz baza danych Microsoft SQL Server Express. Aplikację przygotowano zgodnie ze wzorcem MVC – Model-View-Controller [6]. Stanowiskiem badawczym był laptop o parametrach podanych poniżej:

Procesor Intel Core i5-7300HQ 2.50Ghz, Windows 10 w wersji Home, oraz karta graficzna NVIDIA GeForce GTX 1050 oraz 16 GB RAM.

Każde przeprowadzone badanie składało się z 10 prób. Uzyskane wyniki następnie uśredniano oraz wprowadzono wybrane modyfikacje do zapytań, a badania powtórzono. Na koniec uzyskane wyniki zostały porównane, w celu określenia, która forma zapytania SQL jest bardziej wydajna. Przygotowana metodyka badań wykorzystywała stworzoną aplikację wraz z zapytaniami, które mogły zostać zoptymalizowane, w celu porównania czasów wykonania zoptymalizowanych zapytań SQL z zapytaniami pierwotnymi.

Należy jednak wspomnieć, iż wszystkie badania wykonano w środowisku bez działania dodatkowych usług oraz z wyłączonym programem antywirusowym, tak aby warunki każdego badania były do siebie maksymalnie zbliżone.

W ocenie optymalnej struktury kodu SQL zastosowano kryterium czasu, aby stwierdzić czy wprowadzone do kodu optymalizacje pozwalają jasno określić, kiedy zaproponowana zmiana kodu może być użyteczna.

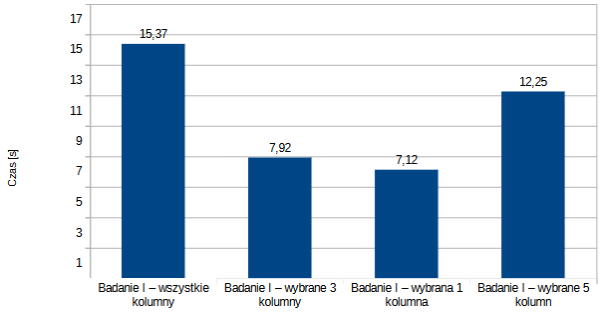
5. Wyniki badań

5.1. Wpływ liczby zwracanych kolumn

Zmniejszenie liczby wybranych kolumn w operacji typu SELECT to proces optymalizacji polegający na ograniczeniu wyboru kolumn tylko do tych koniecznych, w celu skrócenia czasu wykonania zapytania SQL przez silnik bazodanowy.

W badaniu zapytania o różnej strukturze przygotowano cztery próby, odpowiednio wybierając: wszystkie kolumny, pięć kolumn, trzy kolumny oraz finalnie jedną kolumnę.

Na Rysunku 1 przedstawiono wykres dla każdej grupy ze średnimi czasami wykonywania zapytań, które zostały wyrażone w sekundach.



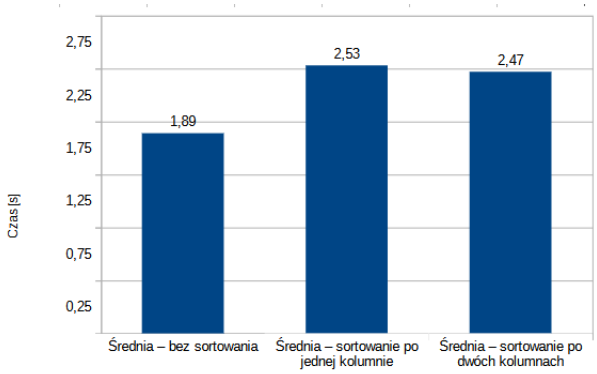
Rysunek 1: Wykres porównawczy uśrednionych czasów zapytań dla różnych ilości kolumn.

Różnica w otrzymanych czasach zapytań jest zauważalna. W powyższym badaniu średni czas wykonania zapytań dla 1000000 rekordów z całej bazy jest najdłuższy (około 2 razy dłuższy od zapytania, w którym wybrana została jedna kolumna) i wynosi 15,37 sekundy. Średni czas zapytania wybierającego tylko jedną kolumnę był najkrótszy i wynosił 7,12 sekundy. Dla trzech kolumn czas wyniósł 7,92 sekundy, zaś dla pięciu kolumn było to 12,25 sekundy. Na podstawie wyników tego badania można stwierdzić, że im większa liczba kolumn w definicji zapytania tym dłuższy czas oczekiwania na wynik jego wykonania.

5.2. Narzut czasowy spowodowany sortowaniem

Usunięcie klauzuli „order by" jest metodą optymalizacji, polegającą na wykluczeniu sortowania danych w kodzie, gdy nie jest ono faktycznie istotne. Silnik bazodanowy w przypadku użycia klauzuli „order by" musi dodatkowo przy zwracaniu wyników posortować dane w odpowiedniej kolejności. W miejscach, w których nie potrzeba posortowanych wyników nie powinno się stosować tej klauzuli.

Na Rysunku 2 przedstawiono wykres ze średnimi czasami wykonywania zapytań, dotyczących klauzuli order by, które zostały wyrażone w sekundach.

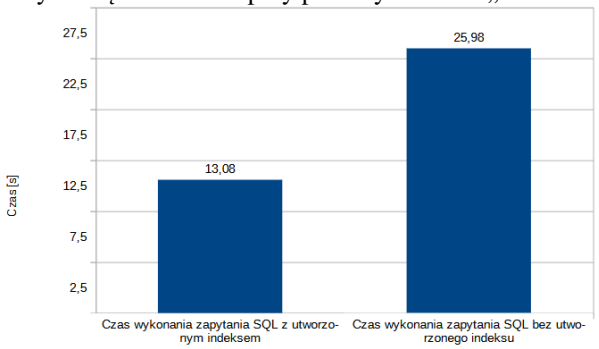


Rysunek 2: Wykres porównawczy uśrednionych czasów zapytań badania II.

Otrzymany wynik w badaniu wpływu klauzuli „order by" na czas zapytania SQL to różnica około 25% pomiędzy czasami realizacji zapytań bez klauzuli „order by", a zapytaniami SQL z tą klauzulą. W przypadku zapytania SQL z klauzulą sortującą istnieje pewność, że wyniki są uporządkowane we wskazanej kolejności. Natomiast w przypadku zapytania SQL bez klauzuli „order by" silnik bazodanowy zwraca nieposortowane wyniki. Jednakże jeżeli nie muszą być one posortowane w odpowiedniej kolejności, można skrócić czas potrzebny na wykonanie zapytania, poprzez brak użycia klauzuli „order by".

5.3. Wpływ indeksowania

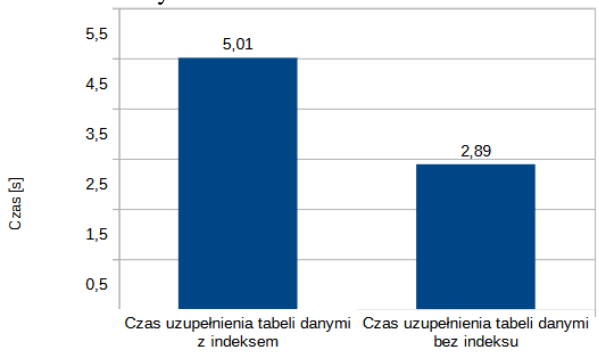
Indeksowanie danych w kolumnach, jest to metoda optymalizacji polegająca na zastosowaniu odpowiedniego indeksu na kolumnie, dzięki której dane są często wyszukiwane. Proces optymalizacji zapytań zaprezentowany w badaniu III polegał na dodaniu indeksu na kolumnie w wybranej tabeli, za pomocą której często odbywa się filtrowanie przy pomocy klauzuli „where".



Rysunek 3: Wykres porównawczy uśrednionych czasów dla zapytań SQL z utworzonym indeksem oraz bez.

Na Rysunku 3 przedstawiono wykres ze średnimi czasami wykonania zapytań z badania wpływu indeksu na czas trwania zapytania SQL. Wartości te zostały wyrażone w sekundach. Średni czas wykonywania zapytania dla polecenia „select", w przypadku gdy nie dodano indeksu wydłużył się o około 13 sekund przy 1000000 rekordów, co w tym przypadku przełożyło się na wzrost o 100% w porównaniu do zapytań, gdzie indeks był utworzony [7].

Jednakże, należy pamiętać o negatywnych aspektach indeksowania. Przy badaniu III zweryfikowano dodatkowo koszt utrzymania indeksów. Zbadano i porównano czasy potrzebne na wstawianie danych do tabeli. Na Rysunku 4 przedstawiono wykres ze średnimi czasami dodawania danych do tabeli.

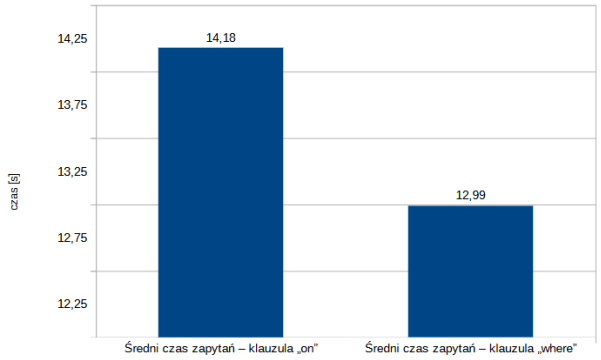


Rysunek 4: Wykres porównawczy czasów dodawania danych do tabeli.

Jak można zauważyć, czas uzupełnienia tabeli milionem wierszy z założonym indeksem jest dłuższy o ponad 2 sekundy niż do tej bez indeksu. Dlatego warto dobrze przemyśleć każdy przypadek z osobna i określić czy indeks jest konieczny.

5.4. Wpływ rodzaju złączenia

Rodzaj złączenia również może wpłynąć na czas trwania zapytania SQL. Badanie czwarte polegało na sprawdzeniu czasów wykonywania zapytań złożonych, w których łączono tabele przy pomocy klauzuli „on" oraz w klauzuli „where". Na Rysunku 5 przedstawiono wykres ze średnimi czasami wykonywania zapytań.



Rysunek 5: Wykres porównawczy uśrednionych czasów zapytań dla złączeń miedzy tabeli w klauzuli „on" oraz „where".

Uśrednione wyniki pomiędzy złączeniami nie są znaczące dla tego badania. Czasy wykonywania zapytań gdy tabele były połączone w klauzuli „on" są dłuższe o 9%, od tych z tabelami połączonymi w klauzuli „where". W tym przypadku, gdy dwie tabele zawierają po 1000000 rekordów jest to różnica około jednej sekundy.

6. Wnioski

Celem artykułu było zbadanie skuteczności zastosowania metod, których zadaniem jest optymalizacja zapytań SQL w środowisku bazodanowym MS SQL. Po przeanalizowaniu dostępnych publikacji naukowych z zakresu optymalizacji zapytań, wybrano cztery różne metody, pozwalające skrócić czas wykonywania zapytań SQL.

Na podstawie wyników badania wpływu liczby kolumn na czas trwania zapytania, można stwierdzić iż teza T1, czyli. „Wybór tylko niezbędnych kolumn w zapytaniu SQL pozwala na zwiększenie wydajności zapytań" jest prawdziwa. Poprzez ograniczenie liczby kolumn w zapytaniu SQL można zaoszczędzić niezbędny czas.

W przypadku tezy T2, tj. „Poprzez usunięcie klauzuli order by nastąpi skrócenie czasu trwania zapytania SQL", drobną zmianą, czyli usunięciem klauzuli można skrócić czas potrzebny na wykonanie zapytania SQL. Warto przejrzeć kod w celu odnalezienia zbędnych klauzuli sortujących.

Wyniki dla badania III potwierdzają postawioną wcześniej tezę T3, tj. „Skrócenie czasu trwania zapytań SQL nastąpi poprzez dodanie odpowiednich indeksów". Dodanie odpowiedniego indeksu pozwoli znacznie skrócić czas potrzebny na wykonanie zapytania „select". Niemniej jednak, należy pamiętać o negatywnym wpływie indeksu w kontekście na przykład dodawania danych do bazy.

Na podstawie wyników badania wpływu rodzaju złączenia tabel, można stwierdzić, iż teza T4, czyli: „Złączenia między tabelami mają wpływ na czasy wykonania zapytania SQL." jest prawdziwa. Można zaobserwować pewną różnicę w czasach realizacji zapytań złożonych SQL, gdy tabele są złączone w klauzulach „on" oraz „where". Skuteczniejszą metodą w tym badaniu okazało się złączenie tabel w klauzuli „where".

Literatura

[1] G. Dziewit, J. Korczyński, M. Skublewska-Paszkowska, Performance analysis of relational databases Oracle and MS SQL based on desktop application, Journal of Computer Sciences Institute 8 (2018) 263-269, https://doi.org/10.35784/jcsi.693.

[2] M. Woch, Optymalizacja Microsoft SQL server przy współpracy z Microsoft Dynamic NAV, Studia Informatica 28(2) (2007) 17-29.

[3] E. Noble, Pro T-SQL 2019, Apress, Berkeley, 2020.

[4] W. Bolesta, Analysis of query execution speed in the selected NoSQL databases, Journal of Computer Sciences Institute 7 (2018) 138-141, https://doi.org/10.35784/jcsi.662.

[5] Otomoto, https://www.otomoto.pl, [01.11.2022].

[6] Model MVC, https://vavatech.pl/technologie/architektura/mvc, [01.11.2022].

[7] SQL - INDEX, https://www.plukasiewicz.net/SQL, [01.11.2022].