

WPLYW MECHANIZMU INDEKSOWANIA DANYCH NA SZYBKOŚĆ REALIZACJI ZAPYTAŃ SQL

Wanda GRYGLEWICZ-KACERKA¹, Jarosław KACERKA²

1. Politechnika Łódzka, Wydział Fizyki Technicznej, Informatyki i Matematyki Stosowanej
tel.: 601 230 683 e-mail: wkacerka@p.lodz.pl
2. Politechnika Łódzka, Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki
tel.: 600 071 801 e-mail: jaroslaw.kacerka@p.lodz.pl

Streszczenie: Przedmiotem pracy jest analiza doświadczalna wpływu wybranych metod indeksowania na czas wykonania polecenia SQL. Badaniom poddano bazę testową wydzieloną z rzeczywistego systemu bankowego. Pomiar czasu wykonania zapytania SQL wykonano dla tabel zawierających do 1000000 rekordów. Badania przeprowadzono dla tabel nie zawierających indeksów oraz tabel zawierających indeksy. Do badań wykorzystano relacyjny system zarządzania bazą danych oparty na Oracle 9. Wyniki badań doświadczalnych pozwoliły na modernizację pracy systemu bankowego.

Słowa kluczowe: indeksowanie, wydajność zapytań SQL, bazy danych.

1. WSTĘP

W każdym projekcie wykorzystującym relacyjną bazę danych, pojawia się problem związany z wydajnością oraz szybkością realizacji zapytań SQL. W celu zmniejszenia czasu wykonywania operacji wyszukiwania danych podczas realizacji zapytań SQL wykorzystuje się indeksy. Indeksy mogą być definiowane na jednym lub kilku atrybutach relacji (zbiorze atrybutów).

W niniejszej pracy badano czas realizacji zapytań SQL. Badania przeprowadzono dla bazy nie zawierającej indeksów oraz dla bazy zawierającej różnego rodzaju indeksy (indeks B-drzewo, bitmapowy, funkcyjny, odwrocony, połączeniowy).

W dokumentacji technicznej Oracle określone są charakterystyczne sytuacje, w których wskazane jest korzystanie z mechanizmu indeksowania:

- indeks powinien zostać stworzony w przypadku, gdy wielokrotnie mamy do czynienia z sytuacją, w której jako wynik zapytań zwracanych jest mniej niż 15 % rekordów pochodzących z dużej tabeli.
- indeksowaniu powinny podlegać kolumny służące do przeprowadzenia złączenia wielu tabel.
- w pewnych okolicznościach zachodzi konieczność założenia indeksów na kluczach głównych, unikalnych lub obcych. Oracle automatycznie tworzy indeksy tylko dla kluczy głównych i unikalnych.

Zaleca się stosować indeksy do kolumn, które charakteryzują się cechą: wartości w kolumnie cechuje unikalność, dane w kolumnie odznaczają się szerokim zakresem wartości, dane zawarte w kolumnie cechuje

niewielki zakres wartości, kolumna zawiera dużą liczbę wartości typu null, ale zapytania dotyczące tych danych w większości przypadków zwracają wszystkie rekordy, które zawierają wartość. Nie należy indeksować: kolumn, które zawierają dane typu LONG RAW i LONG, danych w kolumnie zawierającej dużą ilość wartości null.

System Oracle (w wersji powyżej 10) jest wyposażony w narzędzie analizy czasu wykonywania zapytań SQL. System Oracle 9 nie posiadał jeszcze wielu udogodnień. Wyniki analizy planów zapytań pozwalają odpowiedzieć na pytanie czy zdefiniowany indeks pozwoli na znaczne zwiększenie szybkości wykonywania zapytania. Jeśli odpowiedź jest negatywna to zwykle system Oracle proponuje utworzenie bardziej wydajnego planu wykonania, który to opierałaby się na wykorzystaniu nowego indeksu, którego w bazie jeszcze nie było.

2. BADANIA MODELOWE

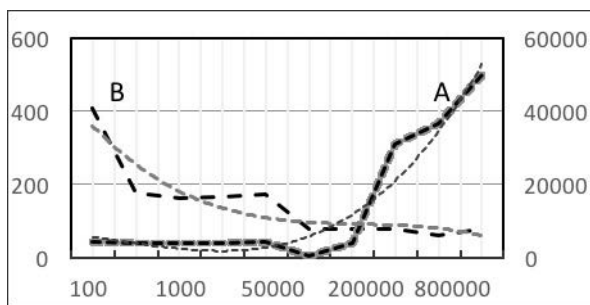
Badania modelowe zostały wykonane dla potrzeb banku, w którym zainstalowano systemem bankowy przeznaczony do kompleksowej obsługi klientów banku - EuroBank On-line (zaprojektowany w Oracle 9). Badania testowe przeprowadzone zostały na zbiorze testowym odzwierciedlającym rzeczywisty system bankowy.

Przyjęto następujące założenia: zbiór testowy składa się z pięciu tabel, użytkownik ma możliwość generowania dowolnej liczności rekordów dla czterech tabel (klienci, pracownicy, produkty, zamówienia), piąta tabela (szczegóły zamówień) stanowi tabelę połączeniową pomiędzy tabelami powiązanymi relacją wiele do wielu. W ramach prowadzenia testów możliwy jest pomiar czasu wykonania wybranego zapytania.

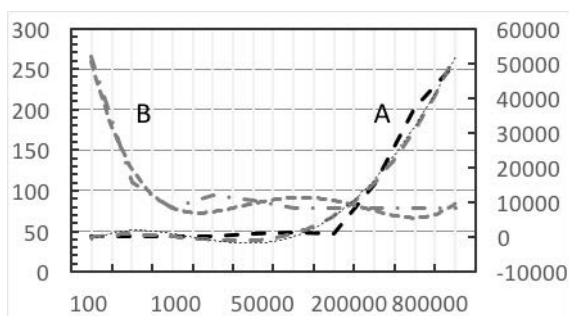
Przeprowadzone badania miały na celu:

- pomiar czasu wykonywania zapytania dla bazy nie zawierającej indeksów,
- pomiar czasu wykonywania zapytania dla bazy z indeksami,
- określenie, które rodzaje indeksów najlepiej sprawdzają się dla określonych typów danych,
- przeanalizowanie czasu wykonywania zapytań dla różnych typów indeksów oraz różnej wielkości tabel.

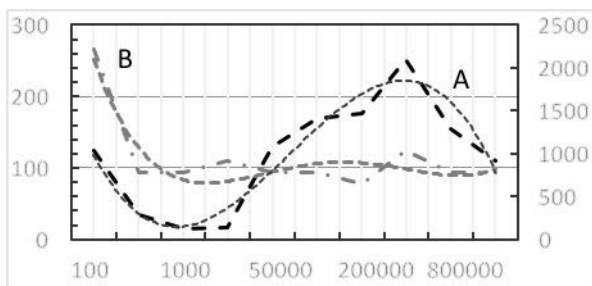
Na rysunkach przedstawione zostały przykładowe wyniki pomiarów (otrzymane z zapytań SQL wyszukujących dane).



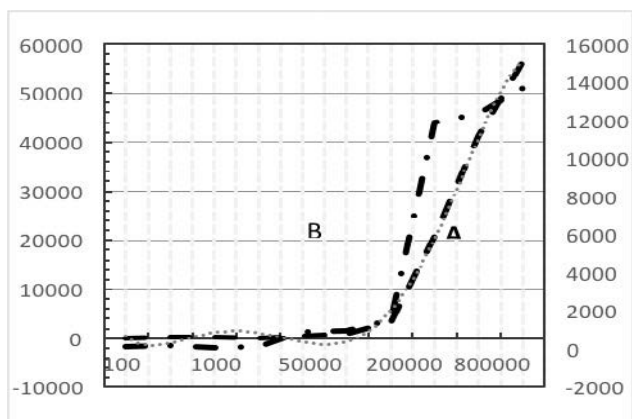
Rys. 1. Czas wykonania zapytania w funkcji ilości rekordów w ms (A-niewykorzystującego indeksu, B-z wykorzystaniem indeksu B-drzewo na jednej kolumnie (dane typu tekst))



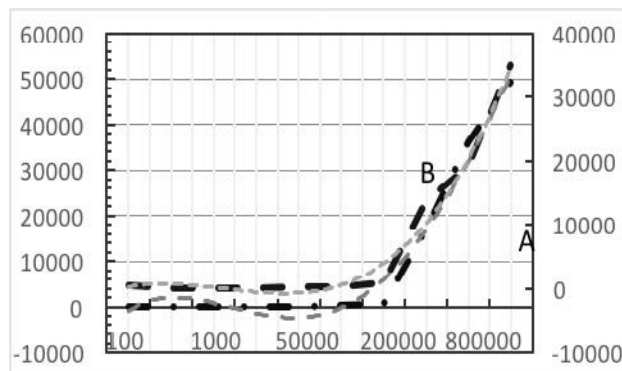
Rys. 2. Czas wykonania zapytania w funkcji ilości rekordów w ms (A-niewykorzystującego indeksu, B-z wykorzystaniem indeksu B-drzewo na dwóch kolumnach (dane typu tekst))



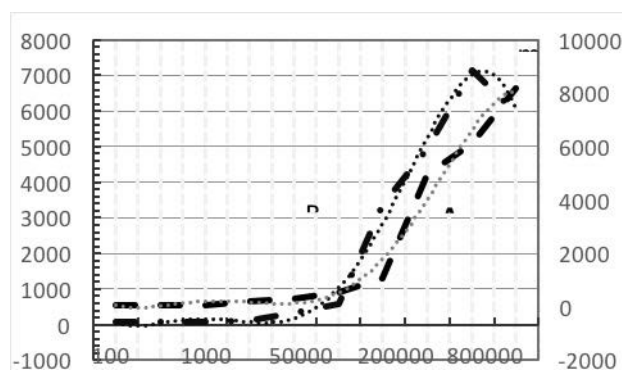
Rys. 3. Czas wykonania zapytania w funkcji ilości rekordów w ms (A-niewykorzystującego indeksu, B-z wykorzystaniem indeksu B-drzewo na jednej kolumnie (dane typu liczba))



Rys. 4. Czas wykonania zapytania w funkcji ilości rekordów w ms (A-niewykorzystującego indeksu, B-z wykorzystaniem indeksu bitmapowego na trzech kolumnach (dane typu tekst))



Rys. 5. Czas wykonania zapytania w funkcji ilości rekordów w ms (A-niewykorzystującego indeksu, B-z wykorzystaniem indeksu funkcyjnego na kolumnach (dane typu tekst))



Rys. 6. Czas wykonania zapytania w funkcji ilości rekordów ms (A-niewykorzystującego indeksu, B-z wykorzystaniem indeksu odwróconego na jednej kolumnie (dane typu tekst))

3. WYNIKI BADAŃ

Otrzymane wyniki badań pozwalają określić skuteczności działania poszczególnych struktur indeksowych (na skrócenie czasu wykonywania zapytania) dla różnej wielkości tabel. W tabeli 1 podano wyniki pomiarów oraz obliczenia.

Z przedstawionej tabeli 1 wynika, że indeks prosty typu B-drzewo, szczególnie dla dużej ilości rekordów, znacznie zmniejszył czas wykonania zapytania (tabela 1, poz. 1, 2, 5, 6) zarówno dla dużej oraz małej liczby wierszy w tabeli.

Indeks złożony typu B-drzewo (założony na dwóch i trzech kolumnach) skutecznie zmniejszył czas odpowiedzi zapytania SQL proporcjonalnie do liczby wierszy w tabeli (tabela 1, poz. 3, 4, 9, 10).

Indeks prosty typu B-drzewo założony na kolumny przechowujące dane w postaci liczbowej, pozwala nieznacznie skrócić czas oczekiwania na odpowiedź w porównaniu z wynikami otrzymanymi dla danych typu tekstowego (tabela 1, poz. 7, 8).

Indeks bitmapowy założony na jednej kolumnie typu tekst w czasie przeprowadzonych badań nieznacznie skrócił czas wykonywania zapytania (tabela 1, poz. 11, 12).

Indeks funkcyjny zdefiniowany na kolumnie tekstowej spowodował nieznaczną poprawę szybkości wykonywania zapytania. Zapytanie wykorzystywało określoną kolumnę jako parametr funkcji.

Tabela 1. Wyniki badań doświadczalnych

(A - tabele bez indeksów, B - tabele z indeksami, D – krotność A/B dla 100 rekordów, E – krotność A/B dla 1000000 rekordów)

l.p.		Liczba rekordów	100	500	1000	10000	50000	100000	200000	500000	1000000	D	E	Rysunek nr.
1	Indeks - tekst	A (ms)	4453	3968	3969	4063	4375	313	3953	30969	36437	10,9	639,8	Rys. 1
2	B-drzewo	B (ms)	407	177	161	166	172	78	78	78	62			
3	Indeks - tekst, tekst	A (ms)	62	156	93	125	812	1266	1000	15391	37000	0,2	631,1	Rys. 2
4	B-drzewo	B (ms)	266	110	78	94	88	78	79	78	78			
5	Indeks - liczba	A (ms)	203	62	109	141	203	563	312	15406	36516	0,5	637,8	Rys. 3
6	B-drzewo	B (ms)	391	110	109	281	601	625	78	81	78			
7	Indeks - liczba, liczba	A (ms)	1032	281	125	141	1078	1406	1469	2094	1296	3,9	9,8	
8	B-drzewo	B (ms)	265	94	93	110	94	94	78	125	93			
9	Indeks - tekst, tekst, liczba	A (ms)	297	110	156	516	1610	1640	1985	20578	41157	0,5	619,0	
10	B-drzewo	B (ms)	547	93	78	79	187	79	79	140	157			
11	Indeks - tekst	A (ms)	125	203	156	140	438	844	3234	20922	41250	0,9	4,1	Rys. 4
12	bitmapowy	B (ms)	140	172	78	141	922	985	1547	11875	12359			
13	Indeks - tekst	A (ms)	78	125	78	109	203	250	813	17125	35922	0,1	1,4	Rys. 5
14	funkcyjny	B (ms)	532	78	79	78	172	235	984	17625	16250			
15	Indeks - tekst	A (ms)	63	78	79	78	234	172	1312	3969	4250	0,8	50,7	Rys. 6
16	odwrócony	B (ms)	79	109	82	156	94	78	78	63	78			
17	Indeks - tekst, liczba	A (ms)	1547	3266	4968	2797	1427	8047	14734	10406	15578	1,0	1,2	
18	bitmapowy połączeniowy	B (ms)	1500	4406	2609	2782	14109	2906	8500	8907	18141			

Nieznaczną poprawę szybkości wykonania zapytania dla tabel o dużej liczności rekordów, natomiast dla małych tabel czas wykonania zapytania nie malał a wręcz wzrósł! (rys. 5).

Indeks odwrócony założony na kolumnę przechowującą dane tekstowe przyspiesza wykonanie zapytania w dość znacznym stopniu dla tabel zawierających bardzo dużą liczbę rekordów. Indeks odwrócony nie nadaje się do indeksacji danych odznaczających się małą różnorodnością i małą selektywnością. Indeks odwrócony jest wykorzystywany do procesu indeksacji sekwencji.

Indeks połączeniowy opiera się na złączeniu tabel. Efekt wzrostu szybkości nieznaczny. Ten typ indeksu tworzony jest dla operacji równościowego połączenia dwóch lub większej ilości relacji, jest on stosowany dla zapytań łączących relacje (tabela 1, poz. 17, 18).

Utworzenie dwóch różnych rodzajów indeksów indeksu B-drzewa i indeksu bitmapowego dla dwóch różnych kolumn tabeli spowodowało niewielki wzrost przyspieszenia wykonywania zapytania dla tabel o niewielkiej liczności (około 50%) i znacznie większy dla tabel o dużej liczności rekordów (około 100%) .

4. WNIOSKI KOŃCOWE

Analiza uzyskanych wyników badań wykazała, że użycie indeksów w większości badanych przypadków pozwala znacznie skrócić czas oczekiwania na wyniki

zapytania SQL, szczególnie dla tabel zawierających duże ilości rekordów. W badanych przypadkach to zmniejszenie czasu było nawet kilkasetkrotne.

Przedstawione wyniki badań doświadczalnych (wykonanych na zbiorze testowym) pozwoliły ocenić różne metody indeksacji danych oraz ich przydatność do skrócenia czasu wykonywania zapytania.

Wyniki badań oraz wnioski wysnute na podstawie przeprowadzonych testów mogą posłużyć administratorom bądź programistom baz danych, którzy stają przed dylematem utworzenie struktur indeksowych w bazie danych. Analiza wyników badań może znacząco pomóc w podjęciu trafnej decyzji doboru struktury indeksowej do zadanej sytuacji.

Pewną trudność, którą udało się pokonać, stanowiło utworzenie uniwersalnego zbioru testowego, który charakteryzował się funkcjonalnością pozwalającą na jednoczesne definiowanie różnych rodzajów struktur indeksowych.

Badania doświadczalne mogą zostać rozszerzone poprzez testowanie innych metod indeksowania dostępnych w Oracle np. B-tree cluster indexes (indeksowanie dzielonych drzew binarnych) oraz domain indexes (indeksy domen).

5. BIBLIOGRAFIA

1. Boroński R., Bocewicz G.: Tables indexing for various densities of SQL queries groups, *Studia Informatica*, Nr 2, Vol. 35, 2014, s. 127-138.
2. Morzy M., Królikowski Z.: *Metody indeksowania atrybutów zawierających zbiory*, Wydawnictwa Pro Dialog NAKOM, Poznań 2012.

IMPACT OF DATA INDEXING ON THE SPEED OF EXECUTION OF SQL QUERIES

The subject of the study is to analyse the impact of selected experimental methods of crawling on the execution time of SQL commands. The study involved test database separate from the actual banking system. Measurements runtime SQL queries made for tables containing up to 1000000 records. The study was conducted for tables that do not contain indexes and tables with indexes. The study used relational database management system based on Oracle. The experimental results have enabled the modernization work of the banking system.

Analysis of the results showed that the use of indices in most of the cases can significantly reduce the waiting time for the results of SQL queries, especially for tables containing a large amount of records. In the cases studied to reduce the time it was even more than 6few hundred times.

Keywords: indexing, query performance SQL database.