STUDIA INFORMATICA 2014
Volume 35 Number 2 (116)

Radosław BOROŃSKI, Grzegorz BOCEWICZ
Politechnika Koszalińska, Wydział Elektroniki i Informatyki

# INDEKSOWANIE TABEL DLA RÓŻNYCH GĘSTOŚCI GRUP ZAPYTAŃ SQL (NA PRZYKŁADZIE ORACLE 11G)

**Streszczenie**. Powszechnie stosowane komercyjne narzędzia doboru indeksów działają na podstawie metod umożliwiających indeksowanie tabel będących częścią niezależnych zapytań SQL. W artykule przedstawiono ideę indeksowania tabel uwzględniającą gęstość grupy zapytań. Przedstawiono wyniki uzyskane autorską Metodą Doboru Indeksów (MDI), opierającą się na algorytmie genetycznym. Przeprowadzone badania pokazują, że dla różnych gęstości zastosowanie indeksu grupowego pozwala skrócić czas wykonania zapytań (o 15%), a także zmniejszyć rozmiar indeksów (o 68-90%).

**Słowa kluczowe**: indeks, indeksowanie, optymalizacja, SQL, SZBD

# TABLES INDEXING FOR VARIOUS DENSITIES OF SQL QUERIES GROUPS (EXAMPLE OF ORACLE 11G)

**Summary**. Commonly used commercial tools are based on a methodology that enables tables indexing for individual SQL queries. The article presents an original method, based on a genetic algorithm, for indexing tables for groups of queries in a relational database. Conducted experiments have shown that the use of indices for a group of queries can reduce the group execution time by 15% as well as can reduce the memory needs by 68-90%.

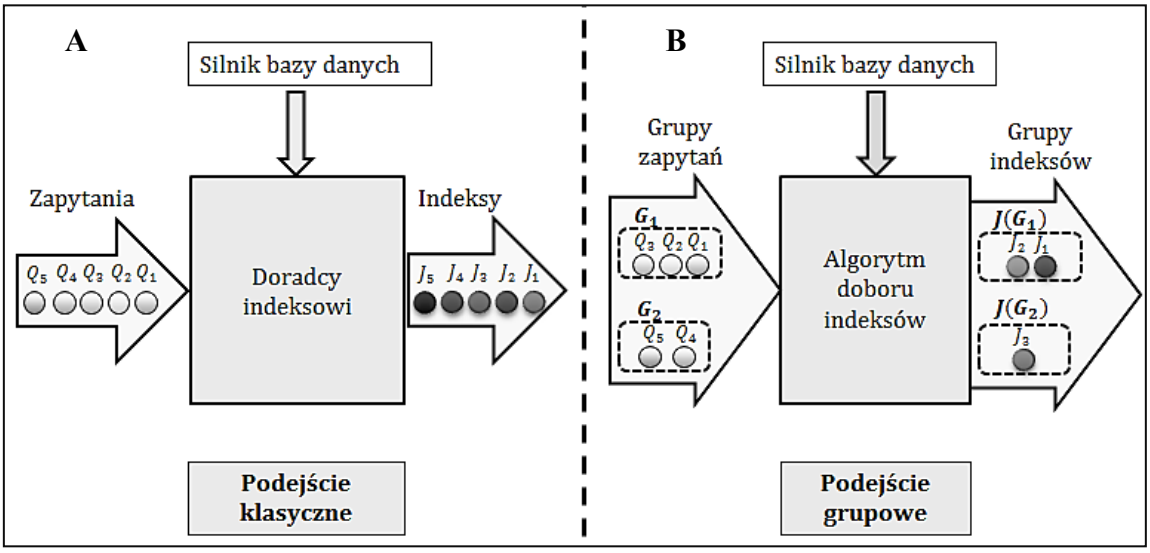
**Keywords**: index, indexing, database, query, optimization, SQL, RDBMS

## 1. Wstęp

Problem doboru indeksów (ang. *Index Selection Problem* – ISP) był już niejednokrotnie omawiany w literaturze [1, 7, 8, 9, 12]. Znane są różne podejścia do optymalnego doboru indeksów dla pojedynczych zapytań [10, 13, 14] i zapytań grupowych [2, 5]. Można do nich

zaliczyć metody opierające się na szacunkach optymalizatora kosztowego, metody operujące na planie wykonania zapytania czy metody analizujące tylko składnię semantyczną zapytania. Pomimo upływu lat problem doboru indeksów jest nadal ważny i aktualny. Chociaż zaproponowane rozwiązania prezentują interesujące próby rozwiązania problemu i mają wiele korzyści (np. indeksowanie większej liczby kolumn, eliminacja powtarzających się struktur [11, 13]), nie są one wystarczające, aby móc zaimplementować je w praktyce [14]. W dotychczasowych opracowaniach zbyt mało uwagi poświęcono licznym niedoskonałościom, do których można zaliczyć: nieuwzględnianie rozmiaru i czasu tworzenia indeksu czy indeksowanie pojedynczego zapytania. Pominięcie innych zapytań odnoszących się do tej samej tabeli może skutkować utworzeniem zbyt dużej liczby podobnych indeksów.

Mając powyższe na uwadze, warto podkreślić, że obecnie dostępne narzędzia komercyjne (np. Oracle Access Advisor, Toad) nie wykorzystują w pełni możliwości indeksowania tabel w ramach grup zapytań. Istnieje potrzeba opracowania automatycznego mechanizmu, który umożliwiłby uwzględnianie związków pomiędzy zapytaniami w grupie i sugerował struktury lepszych indeksów (rys. 1). Celem artykułu jest przedstawienie nowej koncepcji indeksowania tabel w relacyjnych bazach danych. W przeciwieństwie do znanych metod [1, 7, 8], uwzględniających indeksowanie tabel dla pojedynczych zapytań, proponowane rozwiązanie umożliwia wyznaczanie indeksów dla całej grupy zapytań jako całości, dla której poszukiwane są indeksy. Możliwość analizy grupowych zapytań jest szczególnie istotna w bazach danych systemów wspomagania produkcji. Bazy danych tego typu charakteryzują się koniecznością wykonania tysięcy zapytań w ciągu dnia.




Rys. 1. Podejścia doboru indeksów: klasyczne [7, 10] (A), grupowe (B)
Fig. 1. Classic [7, 10] (A) and grouped (B) index selection approaches

W tym kontekście artykuł może być postrzegany jako kontynuacja naszych badań przeprowadzonych w [2, 3, 4, 5], w których podjęto próbę wyznaczenia właściwości grupowych zapytań SQL. W artykule przedstawiono rozszerzaną wersję doboru indeksów. Do rozwiąza-

na tego problemu zaproponowano oryginalną metodę, opierającą się na algorytmie ewolucyjnym, która może być wykorzystana w dowolnej relacyjnej bazie danych.

Rozdział 2 przedstawia sformułowanie problemu doboru indeksów dla relacyjnych baz danych. Rozdział 3 opisuje definicję gęstości zapytań i autorską Metodę Doboru Indeksów (MDI). W rozdziale 4 został przedstawiony plan eksperymentu wraz z wynikami. Rozdział 5 podsumowuje zagadnienie i przedstawia kierunek dalszych badań.

## 2. Sformułowanie problemu

W pracy przyjęto, że problem doboru indeksów jest definiowany następująco.

Dany jest zbiór tabel:

$$T = (T_1,\ldots,T_i,\ldots,T_n) \tag{1}$$

charakteryzowany przez zbiór kolumn wchodzących w skład tych tabel:

$$K = (k_{1,1},\ldots,k_{1,l(1)},\ldots,k_{i,j},\ldots,k_{n,1},\ldots,) \tag{2}$$

gdzie: $k_{i,j}$ – $j$-ta kolumna tabeli $T_i$.

Każdej kolumnie $k_{i,j}$ odpowiada zbiór wartości $V(k_{i,j})$ (zbiór komórek) wchodzących w skład tej kolumny.

Dla zbioru tabel $T$ mogą być formułowane różne zapytania $Q_i$ (w języku SQL są to zapytania typu *SELECT*). Zapytania te są stawiane względem zadanego zbioru kolumn $K^* \subseteq K$. Wynikiem zapytania $Q_i$ jest zbiór:

$$A_i \subseteq \prod_{k_{i,j} \in K^*} V\left(k_{i,j}\right) \tag{3}$$

gdzie: $\prod_{i=1}^{n} Y_i = Y_1 \times Y_2 \times ... \times Y_n$ – iloczyn kartezjański zbiorów $Y_1,\ldots,Y_n$.

Dla zadanej bazy danych *DB* przyjmuje się, że $A_i$ jest wynikiem następującej funkcji:

$$A_i = Q_i\left(K_i^*, Op(DB)\right) \tag{4}$$

gdzie: $K_i^*$ – podzbiór kolumn wykorzystanych w ramach zapytania $Q_i$, $Op(DB)$ – zbiór operatorów dostępnych w bazie *DB*, z których zbudowana jest relacja określająca zapytanie $Q_i$.

Czas związany z wyznaczeniem zbioru $A_i$ jest zależny od wykorzystywanej bazy *DB* (algorytmów przeszukiwania, struktury indeksów itp.) oraz przyjętego zbioru indeksów $J \subseteq P\left(K_i^*\right)$ (gdzie: $P\left(K_i^*\right)$ – zbiór potęgowy $K_i^*$). W ogólnym przypadku przyjmuje się, że czas wykonywania zapytania $Q_i$ w zadanej bazie *DB* jest definiowany jako: $t(Q_i, J, DB)$. Ze względu na to, że rozważania dotyczą wyłącznie wpływu indeksów $J$ (warunki pracy bazy określone w ramach *DB* są uznawane za niezmienne), czas wykonania zapytania $Q_i$ będzie definiowany jako $T_i(J)$.

Zbiór indeksów $J$ oraz tabele wykorzystywane w ramach $Q_i$ są charakteryzowane odpowiednio przez rozmiary wykorzystywanej pamięci $S(J)$ – rozmiar indeksów, $S(Q_i)$ – rozmiar indeksowanych tabel (tabel użytych w ramach $Q_i$). Stosunek tych wielkości określa współczynnik wykorzystania przestrzeni dyskowej indeksów $J$ względem wielkości tabel w zapytaniu $Q_i$: $m(J,Q_i)=\frac{S(J)}{S(Q_i)}$.

W ogólności można rozważać grupę zapytań $Q = \{Q_1, Q_2,\ldots, Q_n\}$ charakteryzowaną przez następujące wielkości:

- zbiór indeksów:

$$J \subseteq P(K^*), \tag{5}$$

- czas wykonywania zapytań całej grupy $Q$:

$$t(J)=\sum_{i=1}^{n} t_i(J), \tag{6}$$

- współczynnik wykorzystania przestrzeni dyskowej:

$$m(J,Q_i)=\frac{S(J)}{\sum_{i=1}^{n} S(Q_i)}. \tag{7}$$

W kontekście tak zdefiniowanych parametrów rozważany **Problem Doboru Indeksów** wiąże się z odpowiedzią na pytanie:

*Jaki zbiór indeksów J, utworzonych dla niezmiennej grupy zapytań Q, spełniających warunek:* $m(J,Q) < r_h$ *(współczynnik wykorzystania przestrzeni nie przekracza zadanej wartości* $r_h$*), minimalizuje czas wykonywania zapytań bazy danych DB:* $t(J) \rightarrow min$*?*

## 3. Metoda Doboru Indeksów

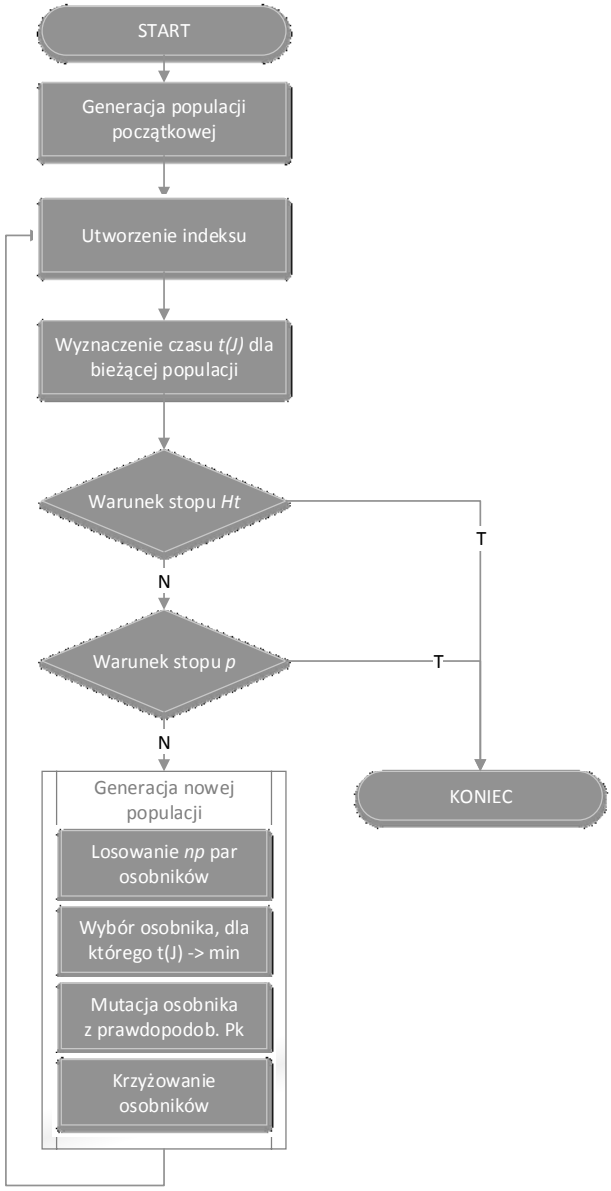
W pracach [3, 4] została przedstawiona metoda rozwiązywania Problemu Doboru Indeksów wykorzystująca ideę algorytmów genetycznych – Metoda Doboru Indeksów. Stanowi ona alternatywę dla metod już istniejących (implementowanych w komercyjnych Doradcach Indeksowych), które w odróżnieniu od proponowanego podejścia nie uwzględniają indywidualnych cech środowiska bazodanowego (np. algorytmu przeszukiwania struktur drzew indeksowych).

Metoda Doboru Indeksów (MDI) dla grupy zapytań $Q$ ma 5 etapów [6]:

- generacja początkowej populacji indeksów, składająca się ze stałej liczby osobników,
- selekcja najlepszych indeksów w populacji,
- mutacja i krzyżowanie indeksów,
- sprawdzenie warunków stopu,
- generacja nowej populacji.

Graficzna ilustracja Metody Doboru Indeksów (MDI) została przedstawiona na rys. 2.

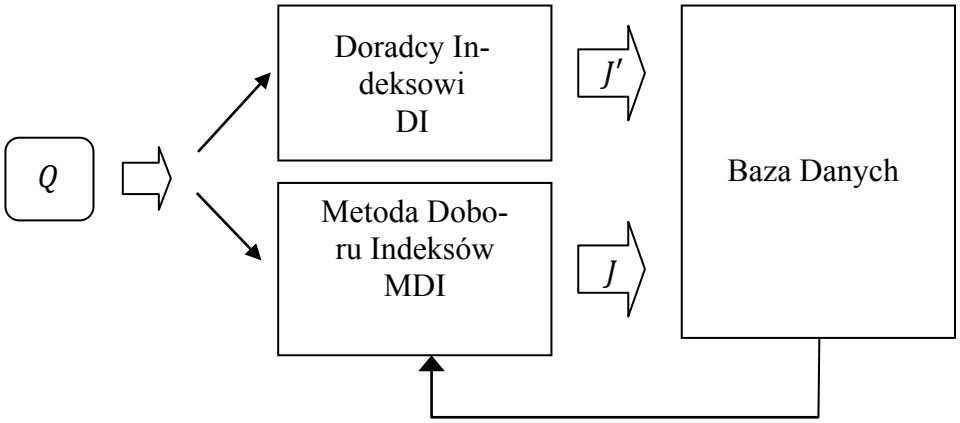
Do zalet stosowania MDI należy zaliczyć możliwość wyznaczania mniejszej liczby indeksów niż po zastosowaniu DI, które raz wyznaczone, mogą być wykorzystywane przy wielu zapytaniach. Wadą metody MDI jest znaczny czas (duża liczba iteracji) potrzebny na uzyskanie zbieżności rozwiązań. W praktyce oznacza to, że aby otrzymać rozwiązanie lepsze niż w przypadku stosowania Doradców Indeksowych, należy uruchomić grupę zapytań wielokrotnie.




Rys. 2. Graficzna ilustracja Metody Doboru Indeksów (MDI) dla grupy zapytań *Q*
Fig. 2. Graphic illustration of Index Selection Method (MDI) for a group of queries *Q*

Na rys. 3 przedstawiono sposób przetwarzania grupy zapytań *Q* przez Doradcę Indeksowego (DI) i Metodę Doboru Indeksów. Grupa zapytań *Q* zostaje poddana działaniu DI tylko

raz, a wypracowane indeksy $J'$ są niezmienne. DI nie potrzebuje dodatkowych iteracji na uzyskanie zbieżnych rozwiązań. Opiera się on na koszcie optymalizatora, analizie planu zapytania lub danych historycznych bazy. W większości przypadków DI znajduje indeksy dla każdego zapytania z grupy $Q$ osobno. Grupa zapytań $Q$ zostaje poddana działaniu MDI kilkakrotnie, a wypracowane indeksy $J$ zmieniają się z każdą iteracją. Oznacza to, że Metoda dostosowuje się do warunków panujących w bazie danych i czasu przetwarzania całej grupy $Q$. Oznacza to, że czas wykonywania grupy zapytań w bieżącej iteracji jest porównywany z czasem z iteracji poprzedniej.




Rys. 3. Przetwarzanie grupy zapytań $Q$ przez DI i MDI
Fig. 3. Grouped queries $Q$ processing by DI and MDI

Istnieją sytuacje, w których proponowana wolniejsza Metoda Doboru Indeksów (MDI) daje te same rezultaty (mierzone jako czas wykonania grupy zapytań $Q$) co szybcy Doradci Indeksowi (DI). Mając to na uwadze, należy określić, przy jakich warunkach zbioru zapytań $Q$ MDI staje się opłacalna, tzn. kiedy indeksy $J$ zaproponowane przez MDI pozwolą wykonać grupę zapytań szybciej niż indeksy ze zbioru indeksów $J'$ (DI). Aby odpowiedzieć na to pytanie, przeprowadzono badania, w których arbitralnie zbiór zapytań $Q$ jest oceniany pod względem tzw. parametru **gęstości grupy zapytań SQL** [2, 3, 4].

Gęstość $\rho_i$ określająca wzajemną relację pomiędzy zapytaniami grupy $Q$ jest liczona jako stosunek liczby kolumn współdzielonych do liczby wszystkich kolumn występujących w grupie zapytań $Q$:

$$\rho_i = \frac{|KW^*|}{|K^*|} \tag{8}$$

gdzie: $\rho_i \in [0,1]$, $\rho_i = 0$ – brak relacji pomiędzy zapytaniami w grupie, $\rho_i = 1$ – występowanie każdej kolumny tabeli w każdym zapytaniu w grupie, $K^*$ – zbiór kolumn tabeli w grupy zapytań $Q$, $KW^* \subseteq K^*$ – podzbiór kolumn występujących co najmniej w dwóch zapytaniach grupy $Q$.

W pracach [2, 3, 4] dostrzeżono, że stosowanie MDI dla niewielkich grup zapytań $Q$ (w których współczynnik gęstości $\rho_i > 0$) pozwala uzyskać indeksy charakteryzujące się krót-

szym czasem przetwarzania grupy $Q$ niż indeksy otrzymane od komercyjnych doradców indeksowych. Korzyść jest tym większa, im większa jest wartość $\rho_i$. W kolejnym rozdziale przedstawiono eksperymenty, które pozwoliły zweryfikować ten warunek w rzeczywistej bazie danych.

## 4. Eksperymenty

Celem przeprowadzonych eksperymentów była weryfikacja skuteczności metody MDI dla grup zapytań $Q$ o strukturach i rozmiarze spotykanych w praktyce.

### 4.1. Środowisko testowe

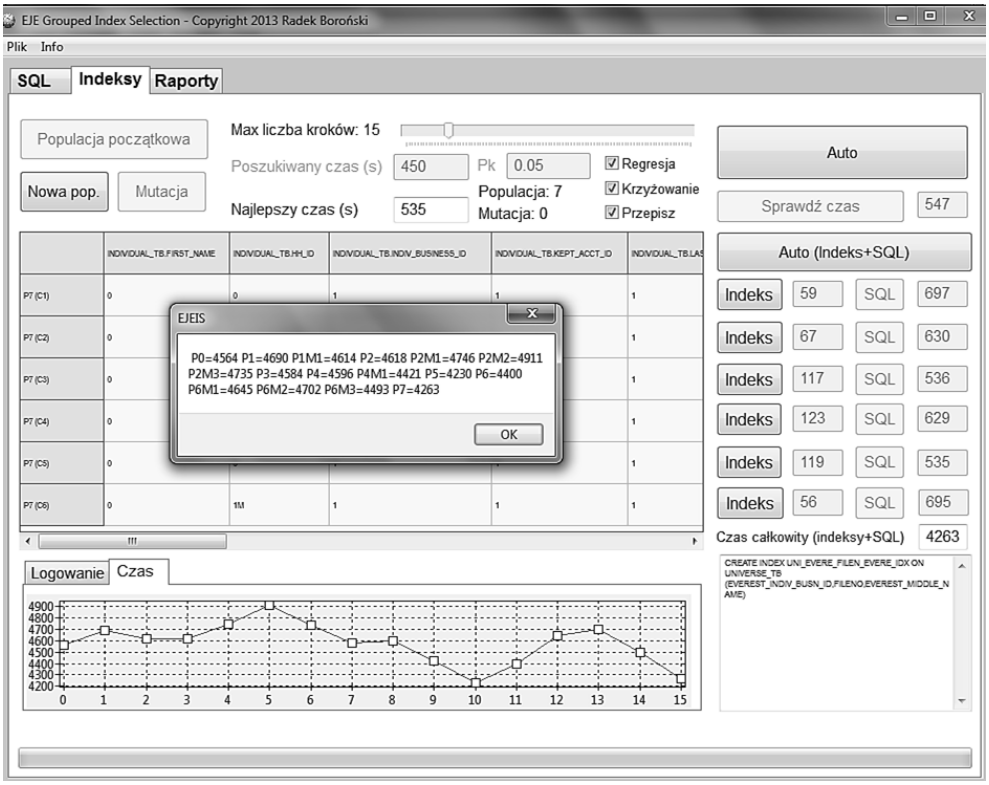
Badania testowe zostały przeprowadzone na dużej (1.2TB) relacyjnej bazie danych Oracle (11.2.0.3) dużego koncernu samochodowego. Na potrzeby testu utworzono kopie 3 tabel zawierających $10^7$ rekordów danych produkcyjnych. Potrzeba utworzenia kopii tabel jest usprawiedliwiona możliwością przypadkowego zablokowania tabeli produkcyjnej w sytuacji jednoczesnego czytania jej przez dwa lub więcej procesów bazodanowych. Na potrzeby eksperymentu został utworzony program komputerowy, napisany w środowisku Lazarus (Free Pascal IDE). Interfejs aplikacji przedstawiono na rys. 4.

Z kilkuset dostępnych w bazie danych zapytań SQL zostały wyselekcjonowane 3 grupy zapytań: $Q^1$, $Q^2$, $Q^3$, które charakteryzowały się czasem przetwarzania krótszym niż czas przetwarzania jednego pełnego cyklu produkcyjnego:

- grupa $Q^1$, składająca się z 31 zapytań, w której są 24 kolumny, z których 11 kolumn występuje w więcej niż jednym zapytaniu,
- grupa $Q^2$, składająca się z 10 zapytań, w której jest 16 kolumn, z których 12 kolumn występuje w więcej niż jednym zapytaniu,
- grupa $Q^3$, składająca się z 41 zapytań, w której jest 39 kolumn, z których 23 kolumny występują w więcej niż jednym zapytaniu.

Cechą charakterystyczną systemu informatycznego generującego zapytania jest to, że wybrane zapytania $Q^i$ są powtarzane cyklicznie. Przykładowo badane zapytania $Q^1$, $Q^2$, $Q^3$ trafiają do bazy raz dziennie.

Doświadczenie i pomiary testowe przeprowadzono dla każdej z grup zapytań $Q^i$ osobno.




Rys. 4. Przykład działania programu dla grupy zapytań $Q^2$
Fig. 4. Example of the computer program runs for a group of queries $Q^2$

### 4.2. Plan eksperymentu

W środowisku opisanym powyżej raz dziennie baza danych otrzymywała zapytania $Q^i_j$ odpowiednie dla swojej grupy $Q^i$.

Grupy zapytań $Q^i$ charakteryzowały się następującą gęstością:

- $\rho_1 = 11/24 = 0{,}46$,
- $\rho_2 = 12/16 = 0{,}75$,
- $\rho_3 = 23/39 = 0{,}59$.

Należy zwrócić uwagę, że w każdej z grup zapytań $Q^i$ został zachowany warunek gęstości $\rho_i > 0$. Oznacza to, że w każdej grupie zapytań $Q^i$ znajdowały się kolumny $K^*_{i,j}$ występujące w więcej niż jednym zapytaniu $Q^i_j$. Można zatem spodziewać się korzyści płynącej z zastosowania metody MDI i jej przewagi nad rozwiązaniami komercyjnymi (DI).

Dla badanych grup zapytań $Q^i$ została przeprowadzona analiza porównawcza. Czasy tworzenia grup indeksów i wykonania grup zapytań dla metody MDI porównano z czasami uzyskanymi w wyniku działania zewnętrznego Doradcy Indeksowego. Do realizacji tego zadania wykorzystano komponent Oracle Access Advisor, będący częścią komercyjnego systemu zarządzania baz danych Oracle 11g. Obydwie metody wypracowywały indeksy, które były następnie porównane. Generowanie jednego kroku (jednego pełnego cyklu) odpowiadało generowaniu jednej populacji w metodzie MDI. Eksperyment był prowadzony przez 60 dni.

Przebadane zostały 3 grupy zapytań $Q^i$. Wyniki eksperymentu przedstawiono w tabelach 1, 2, 3.

Tabela 1

Porównanie wyników dla grupy $Q^1$

| **Grupa $Q^1$** ($\rho_1 = 0{,}46$) ($S(Q_i) = 4075$MB) | **DI** | **MDI** | **Różnica** |
|---|---|---|---|
| Czas wykonania zapytań $t(J)$ | 701 s | 630 s | −71 s |
| Liczba indeksów $\|J\|$ | 1 | 1 | 0 |
| Rozmiar indeksów $S(J)$ | 728 MB | 232 MB | **−496 MB** |
| $m(J,Q_i) = \frac{S(J)}{S(Q_i)}$ | 0,178 | 0,056 | −0,121 |
| Czas potrzebny na uzyskanie wyniku | 20 s | 12 020 s | 12 000 s |

Tabela 2

Porównanie wyników dla grupy $Q^2$

| **Grupa $Q^2$** ($\rho_2 = 0{,}75$) ($S(Q_i) = 19199$MB) | **DI** | **MDI** | **Różnica** |
|---|---|---|---|
| Czas wykonania zapytań $t(J)$ | 627 s | 535 s | **−92 s** |
| Liczba indeksów $\|J\|$ | 4 | 1 | -3 |
| Rozmiar indeksów $S(J)$ | 814 MB | 296 MB | −518 MB |
| $m(J,Q_i) = \frac{S(J)}{S(Q_i)}$ | 0,042 | 0,015 | −0,027 |
| Czas potrzebny na uzyskanie wyniku | 5 s | 54 000 s | 53 995 s |

Tabela 3

Porównanie wyników dla grupy $Q^3$

| **Grupa $Q^3$** ($\rho_3 = 0{,}59$) ($S(Q_i) = 23274$MB) | **DI** | **MDI** | **Różnica** |
|---|---|---|---|
| Czas wykonania zapytań $t(J)$ | 1264 s | 1252 s | −12 s |
| Liczba indeksów $\|J\|$ | 7 | 1 | -6 |
| Rozmiar indeksów $S(J)$ | 2434 MB | 232 MB | **−2202 MB** |
| $m(J,Q_i) = \frac{S(J)}{S(Q_i)}$ | 0,104 | 0,009 | −0,095 |
| Czas potrzebny na uzyskanie wyniku | 64 s | 28 000 s | 27 936 s |

## 5. Podsumowanie

Rozważania teoretyczne odnoszące się do grupy indeksów, zweryfikowane eksperymentalnie na dużej bazie produkcyjnej w rzeczywistych warunkach, przyniosły podobny efekt. Warto zauważyć, że rosnąca gęstość daje lepsze wyniki uzyskane przez MDI w stosunku do DI. Czas potrzebny na analizę grupy zapytań i propozycje indeksów jest dużo krótszy w przypadku doradcy indeksowego w porównaniu z proponowaną metodą. Wynika to ze

sposobu działania doradcy i badania tylko kosztu przetwarzanej grupy zapytań, a nie z faktycznego czasu ich wykonania (cecha MDI). Często analiza taka jest błędna, gdyż opiera się tylko na danych szacunkowych, a nie pomiarowych. Nie są brane pod uwagę istotne elementy przetwarzania bazy danych, takie jak np. struktura fizyczna serwera czy przetwarzanie innej grupy zapytań w tym samym czasie. Wadą podejścia MDI jest liczba dodatkowych iteracji wymaganych do ustabilizowania się populacji i osiągnięcia wartości optymalnych. Niemniej jednak MDI może działać w trakcie cyklicznego wykonywania grupy zapytań, przez co nie powoduje opóźnienia w uzyskaniu wyniku.

Jak to zauważono poprzednio, w wyniku analizy prostych zapytań potwierdzono własność polegającą na tym, że wraz ze wzrostem gęstości zapytań poprawia się również skuteczność tych indeksów wyrażana przez zajmowaną pamięć i czas (np. czas wykonania grupy $Q^2$ lepszy o 15%, rozmiar indeksów $S(J)$ mniejszy o 68% dla grupy $Q^1$, rozmiar indeksów $S(J)$ mniejszy o 90% dla grupy $Q^3$). W przypadku grupy zapytań $Q^1$ zastosowanie wyznaczonego zbioru indeksów pozwala zaoszczędzić w skali roku 33 000 sekund, co w przybliżeniu równa się 53 dniom produkcyjnym. Oszczędność pamięci w rozważanym przypadku jest z kolei na poziomie 90%.

Dalsze badania będą koncentrowały się na ocenie wpływu gęstości zapytań na rozmiar indeksów i czas przetwarzania grupy zapytań.

## BIBLIOGRAFIA

1. Barcucci E., Pinzani R., Sprugnoli R.: Optimal selection of secondary indexes. IEEE Transactions on Software Engineering, Vol. 16(1), 1990, s. 32÷38.
2. Boroński R., Bocewicz G., Wójcik R.: Grouped queries indexing for relational database. eKNOW 2013: The Fifth International Conference on Information, Process, and Knowledge Management, Iaria Journals, 2013, s. 123÷129.
3. Boroński R., Bocewicz G.: Indexes driven mechanism for grouped SQL queries. Pomiary, Automatyka, Robotyka, nr 2, 2013, s. 135÷142.
4. Boroński R.: Problem doboru indeksów dla powiązanych zapytań SQL. [w:] Matuszek J., Gregor M., Micieta B. (red.): Metody i techniki zarządzania w inżynierii produkcji, rocznik VI , Wydawnictwo Naukowe Akademii Techniczno-Humanistycznej w Bielsku-Białej, Bielsko-Biała 2013, s. 27÷42.
5. Boroński R., Bocewicz G.: Multi-criteria index selection for grouped SQL queries. Communications in Computer and Information Science, Vol. 370, Springer, 2013, s. 573÷581.
6. Boroński R., Bocewicz G.: Relational database index selection algorithm (example of Oracle 11g). Communications in Computer and Information Science, Springer, 2014 (in print).

7. Bruno N., Chaudhuri S.: An online approach to physical design tuning. International Conference on Data Engineering, 2007, s. 826÷835.
8. Bruno N., Chaudhuri S.: Automatic physical database tuning: a relaxation-based approach. ACM SIGMOD International Conference on Management of Data, 2005, s. 227÷238.
9. Caprara A., Fischetti M., Maio D.: Exact and approximate algorithms for the index selection problem in physical database design. IEEE Transactions on Knowledge and Data Engineering, Vol. 7(6), 1995, s. 955÷967.
10. Chaudhuri S., Narasayya V.: An efficient Cost-Driven Index Selection Tool for MS SQL Server. Very Large Data Bases Endowment Inc., 1997.
11. Kołaczkowski P., Rybiński H.: Automatic Index Selection in RDBMS by Exploring Query Execution Plan Space. Studies in Computational Intelligence, Vol. 223, Springer, 2009, s. 3÷24.
12. Kratica J., Ljubic I., Tosic D.: A Genetic Algorithm for the Index Selection Problem. EvoWorkshops '03. Proceedings of the 2003 International Conference on Applications of Evolutionary Computing, 2003.
13. Sattler K.-U., Schallehn E., Geist I.: Autonomous query-driven index tuning. International Database Engineering and Applications Symposium, 2004, s. 439÷448.
14. Schnaitter K., Abiteboul S., Milo T., Polyzotis N.: On-line index selection for shifting workloads. International Workshop on Self-Managing Database Systems, 2007, s. 459÷468.

Wpłynęło do Redakcji 30 stycznia 2014 r.

**Abstract**

Finding a good index for a database table is crucial for every relational database system, not only from the productivity point but also because of the cost aspect. Index may be important for a relational database to process queries with reasonable efficiency, but the selection of a good index may be difficult. Presented examples show that there is a need for finding an automatic index selection mechanism for grouped queries. Practice shows that in some circumstances an index focused on grouped queries may give better results and also enables user to save time needed for index creation. It also saves system hardware resources (disk space). In the examples it is shown that grouped queries indexes are more effective than those obtained by classical methods implemented by commercial tools.

**Adresy**

Radosław BOROŃSKI: Politechnika Koszalińska, Wydział Elektroniki i Informatyki, ul. Śniadeckich 2, 75-453 Koszalin, Polska, radoslaw.boronski@tu.koszalin.pl.

Grzegorz BOCEWICZ: Politechnika Koszalińska, Wydział Elektroniki i Informatyki, ul. Śniadeckich 2, 75-453 Koszalin, Polska, bocewicz@ie.tu.koszalin.pl.