

Bożena MAŁYSIAK
Politechnika Śląska, Instytut Informatyki

WARTOŚCI ROZMYTE W ZAGNIEŹDŻONYCH PYTANIACH SQL

Streszczenie. W artykule przedstawiono analizę miejsc występowania wartości rozmytych w zagnieżdżonych pytaniach SQL kierowanych do bazy danych. Wartości rozmyte w zapytaniach zagnieżdżonych mogą pojawić się we wszystkich frazach podzapytania wewnętrznego, we wszystkich frazach podzapytania zewnętrznego lub w warunku wiążącym podzapytanie zewnętrzne z wewnętrznym. Wszystkie rozważane przypadki zostały zilustrowane konkretnymi przykładami.

Słowa kluczowe: rozmyte zagnieżdżone pytania SQL, baza danych, wartości rozmyte

FUZZY VALUES IN NESTED SQL QUERIES

Summary. This article presents analysis place of occurrence of fuzzy values in nested SQL queries. Fuzzy values in nested SQL queries can appear in all phrases of inner subquery, in all phrases of outer subquery or in condition joining outer subquery with inner subquery. All discussed case are illustrated by appropriate examples

Keywords: fuzzy nested SQL queries, database, fuzzy values

1. Wprowadzenie

Artykuł ten powstał w wyniku kontynuacji badań nad miejscami występowania wartości rozmytych w pytaniach SQL [2, 3, 4]. W niniejszym artykule przeprowadzimy taką analizę w odniesieniu do pytań zagnieżdżonych.

Wartości rozmyte w zapytaniach zagnieżdżonych mogą pojawić się w kilku miejscach:

- we wszystkich frazach podzapytania wewnętrznego,
- we wszystkich frazach podzapytania zewnętrznego,
- w warunku wiążącym podzapytanie zewnętrzne z wewnętrznym.

W pytaniach zagnieżdżonych proces wyszukiwania wierszy spełniających rozmyte warunki filtrujące zarówno we frazie *WHERE*, jak i we frazie *HAVING* jest realizowany podobnie jak w pytaniach niezagnieżdżonych [3, 6, 7]. Są jednak pewne różnice:

- pytania zagnieżdżone przyjmują często bardzo skomplikowaną formę,
- pojawia się problem z ulokowaniem warunku na stopień zgodności w przypadku rozmytego warunku wiążącego dwa podzapytania.

Z tych powodów celowe jest szczegółowe rozpatrzenie możliwości umiejscowienia wartości rozmytych w pytaniach zagnieżdżonych.

Ze względu na występowanie wartości rozmytych w warunkach wiążących pytań zagnieżdżonych, wyodrębniono następujący podział:

- 1) porównanie wartości dokładnej podzapytania zewnętrznego z wartością rozmytą (lub zbiorem takich wartości) uzyskiwaną w wyniku wykonania pytania wewnętrznego,
- 2) porównanie wartości rozmytej podzapytania zewnętrznego z wartością dokładną (lub zbiorem takich wartości) uzyskiwaną w wyniku wykonania pytania wewnętrznego,
- 3) porównanie wartości rozmytej podzapytania zewnętrznego z wartością rozmytą (lub zbiorem takich wartości) uzyskiwaną w wyniku wykonania pytania wewnętrznego,
- 4) umiejscowienie wartości rozmytych w warunku korelacji wiążącym skorelowane pytania zagnieżdżone.

Istotnym rozwiązaniem we wcześniejszych badaniach problemem jest sposób zapisu wartości progowej stopnia zgodności dla rozmytego warunku wiążącego dwa podzapytania. Sposób zapisu warunków z wartością progową (przy użyciu formy funkcyjnej i operatorowej) zostanie przedyskutowany w analizowanych dalej przykładach.

Czasami aby skrócić zapis, zamiast *forma operatorowa zapisu warunku na stopień zgodności* będziemy używać krótszego sformułowania *pytanie w formie operatorowej*, podobnie dla formy funkcyjnej.

W celu ułatwienia analizy zaproponowanych pytań przedstawiono fragment bazy danych *ZAKŁADY*, na której opierają się przedstawione przykłady:

```
Instytuty (nr_inst, nazwa)
Zaklad (nr_zakl, nazwa, liczba_pokoi, nr_inst)
Pracownicy (nr_prac, nr_zakl, nazwisko, wiek, plec, staz_pracy)
Zapotrzebowanie (nr_zakl, rok, papier, toner, plytki_CD)
Zuzycie (nr_zakl, rok, papier, toner, plytki_CD)
```

Nazwy kolumn zawierające wartości kluczowe zostały podkreślone, zaś nazwy kolumn zawierające wartości rozmyte pogrubione.

W kolejnych punktach zostaną omówione wszystkie przypadki zapytań wyodrębnionych według podziału ze względu na występowanie wartości rozmytych w warunkach wiążących w pytaniach zagnieżdżonych.

Zapytania zagnieżdżone dzielone są na: **zwykle (nieskorelowane)** i **skorelowane**.

2. Pytania nieskorelowane

W **zwykłych (nieskorelowanych)** pytaniach zagnieżdżonych podzapytanie wewnętrzne wykonywane jest tylko raz, dostarczając w odpowiedzi wiersz lub zbiór wierszy. Następnie dla każdego analizowanego w zapytaniu zewnętrznym wiersza następuje sprawdzenie warunku łączenia z wartością lub zbiorem wartości uzyskiwanych przez pytanie wewnętrzne [11, 12]. Gdy warunek jest spełniony, wiersz pojawia się w zbiorze wynikowym.

2.1. Porównanie wartości dokładnej pytania zewnętrznego z wartością dokładną (lub zbiorem wartości dokładnych) uzyskiwaną przez pytanie wewnętrzne

W tym przypadku warunek wiążący podzapytania ma klasyczną postać, jednakże dopuszczamy występowanie wartości rozmytych zarówno w pytaniu zewnętrznym, jak i wewnętrznym.

Realizacja tego typu pytań jest taka sama, jak pytań w klasycznym języku SQL. Po wyszukaniu wierszy spełniających warunki filtrujące występujące we frazie *WHERE* i *HAVING* pytania zewnętrznego następuje, dla każdego analizowanego wiersza, porównanie wartości kolumny z wartością (lub zbiorem wartości) otrzymaną w wyniku wykonania pytania wewnętrznego. Wiersze spełniające warunki filtrujące pytania zewnętrznego i warunek łączenia tworzą zbiór wynikowy pytania [11, 12].

Przykładowe pytanie kierowane do bazy danych *ZAKŁADY* może brzmieć następująco:

„Wyszukaj te zakłady, które zatrudniają około 30 pracowników i zajmują mniej pokoi niż Zakład Teorii Informatyki. Wyświetl te wiersze, których stopień zgodności wynosi co najmniej 0.55.”

Postać tego pytania zapisana w języku SQL z formą funkcyjną [7] zapisu warunku na stopień zgodności może być wyrażona następująco:

```
SELECT nr_zakl, nazwa
FROM zaklad z JOIN pracownicy p ON z.nr_zakl = p.nr_zakl
WHERE liczba_pokoi < (SELECT liczba_pokoi FROM zaklad
                      WHERE nazwa = 'Zakład Teorii Informatyki')
GROUP BY p.nr_zakl, nazwa
HAVING DG(count(nr_prac) jest okolo 30) >= 0.55;
```

Postać tego pytania zapisana w języku SQL z formą operatorową [7] zapisu warunku na stopień zgodności może być wyrażona następująco:

```
SELECT nr_zakl, nazwa
FROM zaklad z JOIN pracownicy p ON z.nr_zakl = p.nr_zakl
WHERE liczba_pokoi < (SELECT liczba_pokoi FROM zaklad
                      WHERE nazwa = 'Zakład Teorii Informatyki')
GROUP BY p.nr_zakl, nazwa
HAVING DG(count(nr_prac) jest okolo 30) >= 0.55;
```

W podzapytaniu zewnętrznym prezentowanego pytania występuje warunek rozmyty. Po wyznaczeniu stopnia zgodności wartości kolumny analizowanego wiersza z warunkiem pytania oraz porównaniu go z wartością progową, proces wyszukiwania wierszy spełniających warunki zadane w pytaniu zagnieżdżonym realizowany jest w ten sam sposób, co w klasycznych pytaniach zagnieżdżonych języka SQL [11, 12].

W pytaniach, w których w jawny sposób porównuje się stopnie zgodności, wartości rozmyte mogą pośrednio występować w warunku wiążącym podzapytania, na przykład w warunku wyznaczającym stopień zgodności. W tych przypadkach także porównujemy ze sobą wartości dokładne (wartości, jakie przyjmują stopnie zgodności).

Pytanie tego typu może brzmieć następująco:

„Wyszukaj zakłady, w których zapotrzebowanie na rok 2003 na płytki CD oszacowano na około 100 sztuk. W odpowiedzi powinien znaleźć się wiersz najlepiej spełniający kryteria pytania.”

Pytanie to zapisane w języku SQL w formie funkcyjnej może mieć następującą postać:

```
SELECT nr_zakl, nazwa, DG(plytki_CD jest okolo 100) AS stopien
FROM zaklad z JOIN zapotrzebowanie zp
  ON z.nr_zakl = zp.nr_zakl
WHERE zp.rok = '2003' AND DG(plytki_CD jest okolo 100) >= ALL
      (SELECT DG(plytki_CD jest okolo 100)
       FROM zapotrzebowanie zp
       WHERE zp.rok = '2003');
```

Forma operatorowa tego pytania zapisana w języku SQL jest następująca:

```
SELECT nr_zakl, nazwa, (plytki_CD jest okolo 100) AS stopien
FROM zaklad z JOIN zapotrzebowanie zp
  ON z.nr_zakl = zp.nr_zakl
WHERE z.rok = '2003' AND (plytki_CD jest okolo 100) >= ALL
      (SELECT plytki_CD jest okolo 100
       FROM zapotrzebowanie z
       WHERE z.rok = '2003');
```

W tym pytaniu dla każdego analizowanego wiersza pytania zewnętrznego realizowane jest porównanie wartości stopnia zgodności wyznaczonego w wyniku porównania wartości *plytki_CD* z wartością rozmytą *okolo 100* ze zbiorem stopni uzyskanych w wyniku wykonania pytania wewnętrznego.

2.2. Porównanie wartości dokładnej pytania zewnętrznego z wartością rozmytą (lub zbiorem wartości rozmytych) uzyskiwaną przez pytanie wewnętrzne

W tym przypadku głównym problemem, który należało rozwiązać, było:

Jak rozbudować warunek wiążący pytanie zewnętrzne z wewnętrznym o zapis warunku na stopień zgodności?

Przykładowe pytanie kierowane do bazy danych *ZAKŁADY* może brzmieć następująco:

„Wyszukaj zakłady, które w roku 2002 zużyły mniej więcej tyle papieru, ile wynosi średnie oszacowane zamówienie we wszystkich zakładach. W odpowiedzi powinny znaleźć się wiersze o stopniu zgodności co najmniej 0.45.”

Postać tego pytania zapisana w języku SQL w formie funkcyjnej może być następująca:

```
SELECT nr_zakl
FROM zuzycie
WHERE rok = '2002' AND DG(papier mniej_wiecej (SELECT AVG(papier)
FROM zapotrzebowanie
WHERE rok = '2002')) >= 0.45;
```

Funkcja **DG** wyznacza wartość stopnia zgodności wartości kolumny analizowanego wiersza z pytania zewnętrznego do wartości rozmytej uzyskiwanej w wyniku wykonania pytania wewnętrznego.

W pytaniu tym występuje również rozmyta funkcja agregująca *avg()*, której argumentem są wartości rozmyte kolumny *papier* [3, 5].

Pytanie to zapisane w języku SQL w formie operatorowej ma następującą postać:

```
SELECT nr_zakl
FROM zuzycie
WHERE rok = '2002' AND (papier mniej_wiecej (SELECT AVG(papier)
FROM zapotrzebowanie z
WHERE rok = '2002')) >= 0.45;
```

W przypadku gdy w pytaniu konieczne okaże się użycie jednego z operatorów: $<$, $<=$, $>$, $>=$ w warunku wiążącym dwa podzapytania, należy zastosować operacje zgodne z arytmetyką liczb rozmytych L-R [1, 8, 9], tworząc nowe operatory działające na liczbach rozmytych [5, 10, 13].

Jednym ze sposobów określenia, czy wartość dokładna jest w relacji mniejszości lub w relacji większości z wartością rozmytą, jest wykonanie wcześniej rozmycia wartości dokładnej i dopiero wtedy porównanie ze sobą obu wartości.

Należy także zauważyć, że wśród różnych operatorów porównania występujących w warunkach wiążących pytanie zewnętrzne z wewnętrznym postać rozmytą może przyjmować tylko operator równości wyrażony na przykład w postaci operatorów: $\sim=$, *mniej więcej*, *w przybliżeniu* itd [5].

2.3. Porównanie wartości kolumny rozmytej z pytania zewnętrznego z wartością dokładną (lub zbiorem wartości dokładnych) uzyskiwaną przez pytanie wewnętrzne

Proces realizacji takiego pytania przebiega podobnie jak w przypadku przedstawionym w punkcie 2.2.

Przykładowe pytanie kierowane do bazy danych *ZAKŁADY* może brzmieć następująco:

„Wyszukaj zakłady, które w roku 2002 złożyły zapotrzebowanie na liczbę ryz papieru mniej więcej równą średniemu zużyciu we wszystkich zakładach. W odpowiedzi powinny się znaleźć tylko wiersze ze stopniem zgodności co najmniej 0.7.”

Pytanie to zapisane w języku SQL w postaci zagnieżdżonej w formie funkcyjnej może być wyrażone następująco:

```
SELECT nr_zakl
FROM zapotrzebowanie
WHERE rok = '2002' AND DG(papier mniej więcej (SELECT AVG(papier)
                                                FROM zuzycie
                                                WHERE rok = '2002')) >= 0.7;
```

Pytanie to zapisane w języku SQL w postaci zagnieżdżonej w formie operatorowej ma następującą postać:

```
SELECT nr_zakl
FROM zapotrzebowanie
WHERE rok = '2002' AND (papier *>= 0.7) mniej więcej
                        (SELECT AVG(papier)
                         FROM zuzycie
                         WHERE rok = '2002');
```

lub nieco inną bardziej intuicyjną w zapisie formę:

```
SELECT nr_zakl
FROM zapotrzebowanie
WHERE rok = '2002' AND (papier mniej więcej
                        (SELECT AVG(papier)
                         FROM zuzycie
                         WHERE rok = '2002')) >= 0.7;
```

2.4. Porównanie wartości kolumny rozmytej z pytania zewnętrznego z wartością rozmytą (lub zbiorem wartości) uzyskiwaną przez pytanie wewnętrzne

W tym przypadku proces wyznaczania stopnia zgodności i porównywania go z wartością progową realizowany jest tak samo, jak w punktach 2.2 lub 2.3.

Przykładowe pytanie kierowane do bazy danych *ZAKŁADY* może brzmieć następująco:

„Wyszukaj zakłady, które złożyły w roku 2003 zapotrzebowanie na mniej więcej taką liczbę płytek CD jak zakład Oprogramowanie. Wyświetl te wiersze, których stopień zgodności wynosi przynajmniej 0.45.”

Zagnieżdżona postać tego pytania zapisana w języku SQL w formie funkcyjnej może być wyrażona następująco:

```
SELECT nr_zakl
FROM zapotrzebowanie
WHERE rok = '2003' AND DG(plytki_CD mniej_wiecej
    (SELECT plytki_CD
     FROM zapotrzebowanie zp JOIN zaklad z
     ON zp.nr_zakl = z.nr_zakl
     WHERE z.nazwa = 'Zakład Oprogramowania'
     AND zp.rok = '2003')) >= 0.45;
```

Zagnieżdżona postać tego pytania zapisana w języku SQL w formie operatorowej może być wyrażona następująco:

```
SELECT nr_zakl
FROM zapotrzebowanie
WHERE rok = '2003' AND (plytki_CD *>= 0.45) mniej_wiecej
    (SELECT plytki_CD
     FROM zapotrzebowanie zp JOIN zaklad z
     ON z.nr_zakl = z.nr_zakl
     WHERE nazwa = 'Zakład Oprogramowania' AND z.rok = '2003');
```

lub w nieco innej bardziej intuicyjnej w zapisie formie:

```
SELECT nr_zakl
FROM zapotrzebowanie
WHERE rok = '2003' AND (plytki_CD mniej_wiecej
    (SELECT plytki_CD
     FROM zapotrzebowanie zp JOIN zaklad z
     ON zp.nr_zakl = z.nr_zakl
     WHERE nazwa = 'Zakład Oprogramowania' AND z.rok = '2003')) >= 0.45;
```

Sposób konstruowania i działanie operatora `*>` omówione zostaną w punkcie 4 artykułu.

3. Pytania skorelowane

W pytaniach skorelowanych w pytaniu wewnętrznym występuje odwołanie do tablicy (i jej kolumny) zdefiniowanej w pytaniu zewnętrznym.

W pytaniach skorelowanych wartość rozmyta oprócz warunku wiążącego podzapytanie zewnętrzne z wewnętrznym (przypadki te rozważono w poprzednich punktach) może jeszcze pojawić się w warunku korelacji między kolumną aktualnie analizowanego wiersza z pytania zewnętrznego a wartością kolumny analizowanej w pytaniu wewnętrznym.

Taka struktura rozmytych zapytań skorelowanych wymagała przeprowadzenia analizy występowania wartości rozmytych w warunku korelacji oraz właściwej ich interpretacji.

W wyniku przeprowadzonej analizy uznano, że rozwiązanie problemu rozmytej korelacji sprowadzi się do zastosowania metod przedstawionych w artykułach [3, 5, 7], gdyż rozmyta korelacja może być traktowana jak rozmyty warunek filtrujący we frazie *WHERE*, a inna struktura pytań wymusza tylko odpowiednią ich interpretację.

Dla rozpatrywanego warunku korelacji mogą się więc w nim pojawić następujące rodzaje połączeń między wartościami dokładnymi a rozmytymi:

- 1) korelacja wartości dokładnej z pytania zewnętrznego z wartością dokładną pytania wewnętrznego,
- 2) korelacja wartości dokładnej z pytania zewnętrznego z wartością dokładną pytania wewnętrznego (korelacja względem stopnia zgodności),
- 3) korelacja wartości dokładnej z pytania zewnętrznego z wartością rozmytą pytania wewnętrznego lub na odwrót,
- 4) korelacja wartości rozmytej z pytania zewnętrznego z wartością rozmytą pytania wewnętrznego.

3.1. Korelacja wartości dokładnej z pytania zewnętrznego z wartością dokładną pytania wewnętrznego

W pytaniu tym występuje dodatkowo rozmyty warunek wiążący dwa podzapytania.

Przykładowe pytanie kierowane do bazy danych *ZAKŁADY* może brzmieć następująco:

„Wyszukaj zakłady, które zużyły w roku 2002 mniej więcej tyle papieru, ile oszacowały w zapotrzebowaniu. Wyświetl te wiersze, których stopień zgodności przekracza 0.8.”

Zagnieżdżona postać tego pytania zapisana w języku SQL w formie operatorowej może być wyrażona następująco:

```
SELECT nr_zakl
FROM zuzycie zu
WHERE rok = '2002' AND (papier mniej_wiecej
                        (SELECT papier
                         FROM zapotrzebowanie z
                         WHERE rok = '2002'
                         AND z.nr_zakl = zu.nr_zakl)) > 0.8;
```

W zapisie tym warunek korelacji został wyróżniony pogrubioną czcionką.

3.2. Korelacja wartości dokładnej z pytania zewnętrznego z wartością dokładną pytania wewnętrznego (korelacja względem stopnia zgodności)

Przykładowe pytanie kierowane do bazy danych *ZAKŁADY* może brzmieć następująco:

„Wyszukaj te zakłady, których zapotrzebowanie na papier w roku 2003 w tym samym stopniu bliskie jest liczbie około 45.”

Zagnieżdżona postać tego pytania zapisana w języku SQL w formie operatorowej może być wyrażona następująco:

```
SELECT DISTINCT z1.nr_zakl
FROM zapotrzebowanie z1
WHERE z1.rok = '2003' AND EXISTS
      (SELECT * FROM zapotrzebowanie z2
       WHERE z1.nr_zakl <> z2.nr_zakl
        AND z2.rok = '2003' AND
          (z2.papier jest okolo 45) = (z1.papier jest okolo 45));
```

W pytaniu tym w warunku korelacji pośrednio występują wartości rozmyte, gdyż sam warunek operuje już na dokładnych wartościach wyznaczonych stopni zgodności.

3.3. Korelacja wartości dokładnej z pytania zewnętrznego z wartością rozmytą pytania wewnętrznego lub na odwrót

Analizowane w punkcie 3.1 pytanie może zostać zapisane w innej postaci:

„Wyszukaj zakłady, które zużyły w roku 2002 mniej więcej tyle papieru, ile oszacowały w zapotrzebowaniu. W odpowiedzi powinny pojawić się te wiersze, których stopień zgodności wynosi co najmniej 0.6.”

Zapis tego pytania w formie funkcyjnej w języku SQL jest następujący:

```
SELECT nr_zakl
FROM zuzycie zu
WHERE rok = '2002' AND nr_zakl IN
      (SELECT nr_zakl
       FROM zapotrzebowanie z
        WHERE rok = '2002'
        AND DG(z.papier jest okolo zu.papier) >= 0.6);
```

Zapis tego pytania w formie operatorowej w języku SQL może być wyrażony w następujący sposób:

```
SELECT nr_zakl
FROM zuzycie zu
WHERE rok = '2002' AND nr_zakl IN
      (SELECT nr_zakl
       FROM zapotrzebowanie z
        WHERE rok = '2002'
        AND (z.papier jest okolo zu.papier) >= 0.6);
```

3.4. Korelacja wartości rozmytej z pytania zewnętrznego z wartością rozmytą pytania wewnętrznego

Przykładowe pytanie kierowane do bazy danych ZAKŁADY może brzmieć następująco:

„Wyszukaj zakłady, dla których istnieje chociaż jeden inny zakład, który złożył zapotrzebowanie w roku 2003 na podobną liczbę tonerów. W odpowiedzi powinny znaleźć się wiersze ze stopniem zgodności co najmniej 0.9.”

Zagnieżdżone pytanie zapisane w języku SQL w formie funkcyjnej może mieć następującą postać:

```
SELECT nr_zakl
FROM zapotrzebowanie z1
WHERE z1.rok = '2003' AND EXISTS
      (SELECT * FROM zapotrzebowanie z2
       WHERE z2.nr_zakl <> z1.nr_zakl AND
            z2.rok = '2003' AND
            DG(z2.toner jest okolo z1.toner) >= 0.9);
```

Zapis tego pytania w języku SQL w formie operatorowej jest następujący:

```
SELECT nr_zakl
FROM zapotrzebowanie z1
WHERE z1.rok = '2003' AND EXISTS
      (SELECT * FROM zapotrzebowanie z2
       WHERE z2.nr_zakl <> z1.nr_zakl AND
            z2.rok = '2003' AND
            (z2.papier jest okolo z1.papier) >= 0.9);
```

Wszystkie te trzy przypadki, choć w nieco innym aspekcie, zostały rozpatrzone w artykułach [3, 5, 6, 7] podczas analizy rozmytych warunków filtrujących występujących we frazie *WHERE*. Proces porównywania wartości ze sobą, zarówno w przypadku warunku filtrującego, jak i korelacji, realizowany jest w ten sam sposób.

4. Funkcje i operatory realizujące rozmyty warunek wiążący pytanie zewnętrzne z wewnętrznym

Szczególnie trudnym problemem w rozwiązaniach przedstawionych w poprzednich punktach jest uwzględnienie warunku nakładanego na stopień zgodności, towarzyszącego warunkowi rozmytemu.

Przypomnijmy, że powiązanie pytania zewnętrznego z wewnętrznym tworzy warunek filtrujący, który może być zapisany w języku SQL na kilka sposobów [11, 12]:

1. <argument> <op> (<pytanie wewnętrzne>),
2. <argument> <op> {ANY|ALL} (<pytanie wewnętrzne>),
3. <argument> [NOT] IN (<pytanie wewnętrzne>),
4. [NOT] EXISTS (<pytanie wewnętrzne >).

Baza danych utworzona była w SZBD Postgres, wszystkie pytania zadawane do nie realizowane były w języku PostgreSQL.

Pierwszym problemem, który należało tu rozwiązać, to interpretacja takiego warunku, jeśli jeden lub oba argumenty są wartościami rozmytymi.

Stosunkowo prosto problem ten da się rozwiązać w pierwszym przypadku. Po wykryciu przez interpreter PostgreSQL'a niestandardowych typów danych (rozmytych) następuje bowiem wywołanie zewnętrznie zdefiniowanej funkcji obsługującej te typy [5, 10, 13]. Funkcja realizująca operator <op> (wiążący pytanie wewnętrzne z zewnętrznym) została tak skonstruowana, że pozwala na wyznaczenie stopnia zgodności rozmytych argumentów. Uwzględnienie warunku narzuconego na stopień zgodności, tzn. porównanie jego wartości z wartością progową, wykonywane jest wtedy automatycznie przez interpreter PostgreSQL'a [13, 14, 15]. Ilustracją tego przypadku może być następujące pytanie:

„Wyszukać nazwy zakładów, które złożyły maksymalne zapotrzebowanie na papier w roku 2003. W rozwiązaniu uwzględnić wiersze, dla których stopień zgodności przekracza 0.9.”

Zapis tego pytania w opracowanym rozmytym języku SQL ma postać:

```
SELECT nazwa
FROM zaklady z JOIN zapotrzebowanie zp
ON z.nr_zakl = zp.nr_zakl
WHERE zp.rok = '2003' AND (zp.papier ~=(SELECT max(papier)
FROM zapotrzebowanie
WHERE rok = '2003')) > 0.9;
```

Niestety, dla wariantów 2) i 3) system PostgreSQL nie dopuszcza do żadnej modyfikacji zapisu operatorów <op> {ANY| ALL} i [NOT] IN. W rezultacie interpreter zapytań sam inicjuje porównanie lewego argumentu z kolejnymi argumentami zbioru wynikowego pytania wewnętrznego. W trakcie tego porównania niedostępna byłaby wartość progowa (wraz z operatorem) umieszczona w taki sposób, jak w powyższym przykładzie. Dlatego w tym przypadku zastosowano rozwiązanie nie tak eleganckie, jak poprzednio, ale skuteczne, polegające na dołączeniu do prawego lub lewego argumentu rozmytego, wartości progowej wraz z poprzedzającym ją operatorem porównania.

Wracając do analizy zainicjowanego porównania należy wyjaśnić, że interpreter PostgreSQL'a, po stwierdzeniu, że typy argumentów nie są standardowymi typami PostgreSQL, szuka zdefiniowanej przez użytkownika funkcji realizującej operację porównania argumentów i przekazuje jej sterowanie [5, 10, 13, 14, 15]. Funkcja ta operuje na dwóch argumentach rozmytych, wyznacza ich stopień zgodności, a następnie porównuje stopień zgodności z wartością progową.

Na koniec należy jeszcze wyjaśnić, jak dołączyć wartość progową (wraz z operatorem porównania) do jednego z argumentów rozmytych?

Wykorzystano tu możliwość, którą stwarza PostgreSQL, a mianowicie możliwość definiowania własnych operatorów i funkcji je realizujących. Zdefiniowano w tym celu zestaw operatorów o postaci $*\langle op \rangle$, którym towarzyszy wartość progowa, tzn. $*\langle op \rangle \langle wartość\ progowa \rangle$, np. $*\rangle 0.9$.

Zastosowanie takiej konstrukcji przedstawia przykład zapytania:

„Wyszukać nazwy zakładów, które złożyły maksymalne zapotrzebowanie na papier w roku 2003. W rozwiązaniu uwzględnić wiersze, dla których stopień zgodności przekracza 0.9.”

Postać tego pytania, zapisana bez używania funkcji agregującej *max*, w rozbudowanym, rozmytym języku SQL jest następująca:

```
SELECT nazwa
FROM zaklady z JOIN zapotrzebowanie zp
    ON z.nr_zakl = zp.nr_zakl
WHERE zp.rok = '2003' AND zp.papier >= All (SELECT (papier) * >= 0.9
                                         FROM zapotrzebowanie
                                         WHERE rok = '2003');
```

Warunek progowy ≥ 0.9 jest tu dołączany do wyszukanych w pytaniu wewnętrznym wartości rozmytych atrybutu *papier*. Warunek progowy powinien występować w zapisie pytania po wartości rozmytej. W przypadku gdy w warunku wiążącym po jednej i po drugiej stronie występują wartości rozmyte, położenie warunku progowego jest dowolne.

5. Podsumowanie

Artykuł ten powstał w wyniku kontynuacji prac prowadzonych nad miejscami występowania wartości rozmytych w pytaniach SQL, ze szczególnym uwzględnieniem miejsc występowania wartości rozmytych w pytaniach zagnieżdżonych, zarówno nieskorelowanych, jak i skorelowanych. W pracy swą uwagę skupiono przede wszystkim nad konstrukcją i interpretacją rozmytego warunku wiążącego pytanie zewnętrzne z wewnętrznym.

LITERATURA

1. Yager R, Filev D.: Podstawy modelowania i sterowania rozmytego. Wydawnictwa Naukowo-Techniczne, Wiley, Warszawa 1995.
2. Małysiak B.: Mechanizmy wnioskowania przybliżonego w bazach danych. *Studia Informatica*, Vol. 23, nr 4(51). Gliwice 2002.
3. Małysiak B.: Wartości rozmyte w pytaniach SQL do baz danych. *Studia Informatica*, Vol. 24, nr 2A (53). Szczyrk 2003.

4. Małysiak B.: *Aproksymacyjne zapytania do baz danych*. Studia Informatica, Vol. 23, nr 4 (51). Gliwice 2002.
5. Małysiak B.: Tworzenie typów, funkcji i operatorów rozmytych w SZBD Postgres. Konferencja Systemy Czasu Rzeczywistego. Krynica 2002.
6. Małysiak B.: Interpretacja rozmytych warunków filtrujących w zapytaniach SQL. w recenzji
7. Małysiak B.: Wprowadzenie do zapisu zapytań SQL warunku na stopień zgodności. w recenzji.
8. Łachwa A.: *Rozmyty świat zbiorów, liczb, relacji, faktów, reguł i decyzji*. Akademicka Oficyna Wydawnicza Exit, Warszawa 2001.
9. Piegat A.: *Modelowanie i sterowanie rozmyte*. Akademicka Oficyna Wydawnicza Exit, Warszawa 1999.
10. Dybikowski Z.: *PostgreSQL*, Helion, Gliwice 2001.
11. Ullman J. D., Widom Z.: *Podstawowy wykład z systemów baz danych*. Wydawnictwa Naukowo-Techniczne, Warszawa 2000.
12. *SQL Język Relacyjnych Baz Danych*. Wellesley Software. Wydawnictwa Naukowo-Techniczne, Warszawa 1995.
13. *PostgreSQL 7.2 Programmer's Guide*. The PostgreSQL Global Development Group.
14. *PostgreSQL 7.2 Reference Manual*. The PostgreSQL Global Development Group.
15. *PostgreSQL 7.2 User's Guide*. The PostgreSQL Global Development Group.

Recenzent: Dr hab. inż. Stanisław Wołek Prof. Pol. Rzeszowskiej

Wpłynęło do Redakcji 19 lutego 2004 r.

Abstract

This article presents analysis place of occurrence of fuzzy values in nested SQL queries. Fuzzy values in nested SQL queries can appear in:

- all phrases of inner subquery,
- in all phrases of outer subquery
- or in condition joining outer subquery with inner subquery.

The first chapter presents short introduction about nested SQL queries.

In the second chapter all types of nested incorrelated queries are discussed. All of these types are illustrated by appropriate examples.

In the third chapter all types of nested correlated queries are described. All of these types are illustrated by appropriate examples.

The fourth chapter is about functions and operators which realise fuzzy condition joining outer subquery with outer subquery.

The article is ended by short resume.

Adres

Bożena MAŁYSIAK: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,
44-101 Gliwice, Polska, bozena@ivp.iinf.polsl.gliwice.pl .