

Bożena MAŁYSIAK
Politechnika Śląska, Instytut Informatyki

UZUPEŁNIANIE ZAPYTAŃ SQL O STOPIEŃ ZGODNOŚCI KRYTERIUM WYBORU

Streszczenie. W artykule przedstawiono kilka możliwych form wprowadzania do zapytań SQL jawnego zapisu warunku na wartość stopnia zgodności. Zarówno dla prostych, jak i zagnieżdżonych pytań SQL omówiono trzy formy zapisu warunku na stopień zgodności kryterium wyboru: dołączaną, funkcyjną i operatorową. Formy te zilustrowano przykładami.

Słowa kluczowe: zapytania SQL, stopień zgodności, kryterium wyboru

COMPLEMENTATION SQL QUERIES ON DEGREE OF COMPATIBILITY OF CRITERION OF SELECTION

Summary. This article presents several possible forms of introduction to SQL queries notation of condition on degree of compatibility. In this work for both simple and nested SQL queries three forms notation of condition on degree of compatibility are discussed: joined, functioned and operator. These forms are illustrated by examples.

Keywords: SQL queries, degree of compatibility, criterion of selection

1. Wprowadzenie

Wprowadzenie do pytań SQL-owych warunków rozmytych prowadzi w procesie interpretacji pytań do wyznaczenia stopnia zgodności wszystkich wyszukiwanych wierszy z kryterium pytania [4, 5, 7]. Powstaje wtedy problem końcowego wyboru wierszy, które mają się znaleźć w odpowiedzi [10]. Można tu wymienić kilka metod:

- wybranie wierszy o maksymalnym stopniu zgodności z warunkami pytania,
- wybranie wierszy o stopniu zgodności przekraczającym średni stopień,

- wybranie wierszy o stopniu zgodności przekraczającym pewną zadaną wartość progową.

W niniejszym artykule rozważymy najciekawszy, trzeci z wymienionych przypadków. Wymaga on rozwiązania sposobu wprowadzania takiej wartości progowej.

Wartość progowa stopnia zgodności może zostać wprowadzona do systemu poprzez:

- a) wstępne przyjęcie stałej wartości progowej,
- b) jawne ustalenie warunku na stopień zgodności w pytaniu SQL.

Wstępne przyjęcie wartości progowej jest prostsze w implementacji, ale ogranicza w pewnym stopniu elastyczność konstruowania zapytań. Można bowiem rozważać przypadki złożonych zapytań SQL-owych, zwłaszcza zagnieżdżonych, gdzie warunki rozmyte pojawiają się w różnych frazach zapytań i zajdzie potrzeba uwzględnienia dla różnych fraz różnych stopni zgodności z kryteriami pytania [11]. Dlatego w tym artykule rozważymy kilka możliwych form wprowadzenia do zapytań SQL jawnego zapisu warunku na wartość stopnia zgodności.

2. Formy zapisu warunku na wartość stopnia zgodności

Rozważmy trzy, arbitralnie przyjęte, formy zapisu warunku na stopień zgodności:

- forma dołączana do warunku rozmytego,
- forma funkcyjna,
- forma operatorowa.

Formy te zilustrujemy przykładem następującego pytania:

„Wyszukaj pracowników, którzy mają około 50 lat, a w odpowiedzi uwzględnij wiersze ze stopniem zgodności z kryterium pytania większym niż 0.7”

Dotychczasowy zapis tego pytania bez uwzględniania warunku na stopień zgodności, przedstawiał się następująco [4, 6, 7]:

```
SELECT nazwisko
FROM pracownicy
WHERE wiek jest okolo 50;
```

2.1. Forma dołączana do warunku rozmytego

Do zapisu warunku na stopień zgodności w tej formie wprowadzimy predefiniowaną zmienną DG (*ang. degree*). Warunek nakładany na stopień zgodności jest wtedy dołączany (poprzez koniunkcję) do zapisu warunku rozmytego we frazie *WHERE*:

```
WHERE wiek jest okolo 50 AND DG > 0.7;
```

Rozwiązanie to wymaga jednak kontekstowej analizy całego warunku, co jest trudne w implementacji.

W formalnym zapisie warunku rozmytego w tej formie wykorzystamy definicję złożonego warunku rozmytego przedstawioną w artykule [7].

Warunek ten jest zdefiniowany następująco:

```

<war_rozmyty_złożony> ::= <war_rozmyty_prosty> |
                        <war_rozmyty_złożony><funktor><war_rozmyty_prosty>
<war_rozmyty_prosty> ::= <argument><operator><argument>
<argument> ::= <wartość_dokładna> | <wartość_rozmyta>
<operator> ::= ~ = | > | < | <= | >= | <>
<funktor> ::= AND | OR | AND NOT | OR NOT
...
<war_rozszerzony> ::= <war_rozmyty_złożony> AND <war_progowy>
<war_progowy> ::= DG<operator_rel><wartość_progowa>
<operator_rel> ::= = | < | > | <= | >= | <>
<wartość_progowa> ::= <liczba dziesiętna z przedziału [0, 1]>
...

```

2.2. Forma funkcyjna

W celu powiązania warunku na stopień zgodności z warunkiem rozmytym zdefiniowano funkcję DG, której użycie w rozważanym przykładzie ma następującą postać:

```
WHERE DG(wiek jest okolo 50) > 0.7
```

Rozpatrywano też alternatywną formę, którą ilustruje zapis:

```
DG(p.wiek, okolo 50) > 0.7;
```

Formalny zapis rozszerzonej pierwszej z przytoczonych postaci warunku filtrującego z wykorzystaniem tej funkcji przedstawia się następująco:

```

...
//wykorzystamy definicję złożonego warunku rozmytego przedstawioną
w punkcie 2.1
<war_rozszerzony> ::= DG(<war_rozmyty_złożony>)<war_progowy>
<war_progowy> ::= <operator_rel><wartość_progowa>
<operator_rel> ::= = | < | > | <= | >= | <>
<wartość_progowa> ::= <liczba dziesiętna z przedziału [0, 1]>
...

```

2.3. Forma operatorowa

Trzecia z rozpatrzonych form zapisu jest uproszczeniem formy funkcyjnej. Wykorzystanie jej do rozpatrywanego przykładu wygląda następująco:

```
WHERE (wiek jest okolo 50) > 0.7;
```

Interpretacja warunku rozmytego w powyższym zapisie prowadzi do wyznaczenia stopnia zgodności, który jest natychmiast porównywany z wartością progową 0.7. Formalny zapis rozszerzonej postaci warunku filtrującego dla tego przypadku jest następujący:

```

...
//wykorzystamy definicję złożonego warunku rozmytego przedstawioną
w punkcie 2.1
<war_rozszerzony> ::= (<war_rozmyty_złożony>)<war_progowy>
<war_progowy> ::= <operator_rel><wartość_progowa>
<operator_rel> ::= = | < | > | <= | >= | <>
<wartość_progowa> ::= liczba dziesiętna z przedziału <0, 1>
...

```

3. Zastosowanie wprowadzonych form zapisu na stopień zgodności w niezagnieżdżonych pytaniach SQL

Przykłady zapytań w języku SQL oparte są na bazie danych *ZAKŁADY*. W skład bazy danych *ZAKŁADY* wchodzi między innymi następujące tabele:

```

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 wchodzących w skład kluczy głównych w tabelach zostały podkreślone, natomiast nazwy kolumn zawierających wartości rozmyte zostały pogrubione.

Przechowywane w tabeli *Zapotrzebowanie* dane dotyczą szacowanych (planowanych) potrzeb na dany rok.

W przedstawionych przykładach rozpatrywane warunki rozmyte będą, dla wyróżnienia, pogrubione.

3.1. Pojedynczy warunek filtrujący umieszczony we frazie *WHERE*

Rozważmy na początku pytanie zawierające jeden rozmyty warunek filtrujący umieszczony we frazie *WHERE*:

```
WHERE <warunek filtrujący>;
```

3.1.1. Warunek nałożony na kolumnę zawierającą wartości dokładne

Przykładowe pytanie tej postaci zadawane do bazy danych *ZAKŁADY* może brzmieć następująco:

„Wyszukaj zakłady, w których pracuje około 30 pracowników. W odpowiedzi powinny się znaleźć wiersze ze stopniem zgodności z kryterium pytania równym lub większym niż 0.8.”

Pytanie takie może zostać zapisane w języku SQL z funkcyjną formą zapisu warunku na stopień zgodności w następujący sposób:

```
SELECT z.nr_zakl
FROM zaklad z JOIN pracownicy p ON z.nr_zakl = p.nr_zakl
GROUP BY z.nr_zakl
HAVING DG(count(nr_prac) jest okolo 30)>= 0.8;
```

Postać tego pytania z operatorową formą zapisu warunku na stopień zgodności jest następująca:

```
SELECT z.nr_zakl
FROM zaklad z JOIN pracownicy p ON z.nr_zakl = p.nr_zakl
GROUP BY z.nr_zakl
HAVING (count(nr_prac) jest okolo 30)>= 0.8;
```

Realizację tego typu pytań bez wartości progowej stopnia zgodności przedstawiono w artykule [7]. W powyższym pytaniu we frazie *WHERE* występuje wartość rozmyta *okolo 10*. Ogólnie można stwierdzić, że dla każdej grupy wierszy tabeli wyznaczany jest stopień zgodności z warunkiem `count(nr_prac) jest okolo 10`. Następnie wybierane są te grupy, których stopień zgodności jest równy lub większy niż 0.8.

3.1.2. Warunek nałożony na kolumnę zawierającą wartości rozmyte

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

„Wyszukaj zakłady, których zapotrzebowanie na papier w roku 2003 wynosiło około 45 ryz, zaś stopień zgodności z tym warunkiem nie jest mniejszy od 0.6”

Pytanie takie może zostać zapisane w języku SQL w formie operatorowej w następujący sposób:

```
SELECT nr_zakl
FROM zapotrzebowanie
WHERE rok = '2003' AND (papier jest okolo 45) >= 0.6;
```

We frazie *WHERE* tego pytania występuje porównanie wartości rozmytej *okolo 45* z wartościami rozmytymi kolumny *papier* i wyznaczenie stopnia zgodności. Każda z tych wartości reprezentowana jest przez zdefiniowaną wcześniej funkcję przynależności.

Na kolumnę rozmytą może być również nakładany dokładny warunek filtrujący. Wyznaczanie stopnia przynależności dla takiego pytania przebiega w ten sam sposób, co w przypadku nakładania warunku rozmytego na kolumnę zawierającą dokładne dane, proces ten został opisany w punkcie 3.1.1.

Przy nakładaniu warunków filtrujących na kolumny zawierające wartości rozmyte można wykorzystywać także inne operatory relacyjne, takie jak: $<$, $>$, \geq , \leq . W tym przypadku wartości rozmyte należy potraktować jako liczby rozmyte (najwygodniej

przedstawione w postaci $L-R$ [1, 8, 9]. Określenie, która z liczb jest większa, może polegać na przykład na porównaniu ich wartości modalnych, a w przypadku przedziałów rozmytych na przykład na porównaniu wartości największych, dla których funkcje przynależności przyjmują jeszcze wartość 1.

3.2. Nakładanie różnej wartości progowej dla każdego warunku oddzielnie

Jeśli we frazie *WHERE* wystąpi kilka warunków filtrujących, można dla każdego z nich wprowadzić oddzielny warunek na stopień zgodności z inną wartością progową. Wtedy dla każdego warunku oddzielnie należy określić, czy jego stopień zgodności jest równy lub przekracza wartość progową czy nie. W tym przypadku warunki filtrujące mogą być połączone spójnikami logicznymi (*AND*, *OR*).

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

„Wyszukaj zakłady, które zajmują około 15 pokoi ze stopniem zgodności nie mniejszym od 0.5 i których zapotrzebowanie na papier wyniosło 15 ryz ze stopniem zgodności nie mniejszym od 0.8 lub 4 tonery ze stopniem zgodności przekraczającym 0.75.”

Pytanie to zapisane w języku SQL w formie operatorowej może mieć następującą postać [4]:

```
SELECT nr_zakl
FROM zaklad z JOIN zapotrzebowanie zp
ON z.nr_zakl = zp.nr_zakl
WHERE (liczba_pokoi jest okolo 15) >= 0.5 AND
      ((zp.papier jest 15) = 0.8 OR (zp.toner jest 4) > 0.75);
```

3.3. Warunek filtrujący złożony z kilku warunków dokładnych i rozmytych połączonych różnymi funktorami

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

„Wyszukaj te zakłady, które zajmują około 15 pokoi i których zeszłoroczne zapotrzebowanie na tonery lub papier mniej więcej pokryło się ze zużyciem. Wyświetl te wiersze, których stopień zgodności przekracza 0.8.”

Postać tego pytania w języku SQL w formie operatorowej może być wyrażona następująco:

```
SELECT nr_zakl
FROM zaklad z JOIN zapotrzebowanie zp
ON z.nr_zakl = zp.nr_zakl
JOIN zuzycie zu ON zu.nr_zakl = zp.nr_zakl
WHERE (liczba_pokoi jest okolo 15 AND
      zu.rok = '2003' AND z.rok = '2003' AND
      (zu.toner jest zp.toner) OR (zu.papier jest zp.papier)) > 0.8;
```

Realizacja tego typu pytań wykorzystuje te same reguły, które były opisane artykułach [2, 3, 4, 7]. Analogicznie do punktów 1, 2 i 3 z punktu 3.1 oddzielamy warunki dokładne i rozmyte. Adekwatnie do zastosowanego funktora wyznaczamy stopień zgodności. Na końcu sprawdzany jest warunek na stopień zgodności.

4. Zapis wartości progowej stopnia zgodności w pytaniach zagnieżdżonych

Istotnym rozwiązaniem w badaniach problemem jest sposób zapisu wartości progowej stopnia zgodności dla rozmytego warunku wiążącego dwa podzapytania.

Podobnie jak w pytaniach niezagnieżdżonych, można zaproponować trzy formy takiego zapisu. Formalny zapis warunku wiążącego dwa podzapytania, w tych formach, jest zdefiniowany następująco:

– Forma dołączana:

```
...
<forma_dolaczana_w_w> ::= <wyrażenie_pył_zewn><op>(<pytanie_wewn>)
                        AND DG<warunek_progowy>
<op> ::= <operator_rel>| <operator_rel><operator_zb>| IN| NOT IN
<warunek_progowy> ::= <operator_rel><wartość_progowa>
<operator_rel> ::= = | < | > | <= | >= | <>
<wartość_progowa> ::= liczba dziesiętna z przedziału <0, 1>
<operator_zb> ::= ANY| ALL
...
```

– Forma funkcyjna:

```
...
<forma_funkcyjna_w_w> ::=
                        DG(<wyrażenie_pył_zewn><op>(<pytanie_wewn>))
                        <warunek_progowy>
<op> ::= <operator_rel>| <operator_rel><operator_zb>| IN| NOT IN
<warunek_progowy> ::= <operator_rel><wartość_progowa>
<operator_rel> ::= = | < | > | <= | >= | <>
<wartość_progowa> ::= liczba dziesiętna z przedziału <0, 1>
<operator_zb> ::= ANY| ALL
...
```

– Forma operatorowa:

```
...
<forma_operatorowa_w_w> ::=
                        <wyrażenie_pył_zewn><op_r>(<pytanie_wewn>)
                        <warunek_progowy> |
                        <wyrażenie_pył_zewn><op>(<pytanie_wewn_rozsz_r>) |
```

```

        <wyrażenie_rozmyte_rozsz><op>(<pytanie_wewn>)
<pytanie_wewn_rozsz_r> ::=
        SELECT <wyrażenie><op_rozsz>
        <wartość_progowa><kolejne_frazy_pytania_wewn>
<wartość_rozmyta_rozsz> ::= <wartość_rozmyta><op_rozsz>
        <wartość_progowa>
<op_rozsz> ::= *= | *> | *>= | *< | *<= | *<
<op> ::= <op_rel>| <op_rel><op_zb>| IN| NOT IN
<op_r> ::= ~=
<warunek_progowy> ::= <op_rel><wartość_progowa>
<op_rel> ::= = | < | > | <= | >= | <>
<wartość_progowa> ::= liczba dziesiętna z przedziału <0, 1>
<operator_zb> ::= ANY| ALL
...

```

Sposób zapisu warunku łączącego w pytaniach zagnieżdżonych z wartością progową (przy użyciu formy operatorowej) zostanie przedstawiony na przykładzie.

„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.”

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;

```

5. Podsumowanie

W niniejszym artykule omówiono sposoby wprowadzania do zapisu zapytań SQL warunku na stopień zgodności zarówno w prostych, jak i zagnieżdżonych pytaniach SQL. Celowość rozpatrywania trzech różnych form zapisu uzasadnić można następująco:

- forma dołączana jest formą najprostszą, jednak jej zastosowanie jest ograniczone do prostych pytań niezagnieżdżonych,
- forma funkcyjna jest bardzo elastyczna i daje przejrzysty zapis nawet dla pytań zagnieżdżonych. Jednak ograniczenia implementacyjne powodują, że dla pewnych specyficznych pytań zagnieżdżonych jest ona niewystarczająca,
- forma operatorowa jest najbardziej uniwersalna, daje bardzo zwarty zapis, choć nie zawsze tak czytelny, jak dwie pierwsze formy.

LITERATURA

1. Yager R, Filev D.: Podstawy modelowania i sterowania rozmytego. Wydawnictwa Naukowo-Techniczne, Wiley. Warszawa 1995.
2. Yu C.T., Meng W.: Principles of Database Query Processing for Advanced Applications. Morgan Kaufmann Publishers, Inc., 1998.
3. Małysiak B.: Mechanizmy wnioskowania przybliżonego w bazach danych. *Studia Informatica*, Vol. 23, nr 4(51). Gliwice 2002.
4. Małysiak B.: *Wartości rozmyte w pytaniach sql do baz danych*. *Studia Informatica*, Vol. 24, nr 2A (53). Szczyrk 2003.
5. Małysiak B.: *Aproksymacyjne zapytania do baz danych*. *Studia Informatica*, Vol. 23, nr 4 (51). Gliwice 2002.
6. Małysiak B.: Tworzenie typów, funkcji i operatorów rozmytych w SZBD Postgres. Konferencja Systemy Czasu Rzeczywistego. Krynica 2002.
7. Małysiak B.: Interpretacja rozmytych warunków filtrujących w zapytaniach SQL. 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. Badurek J.: *Logika rozmyta w bazach danych*. Informatyka. Styczeń 1999.
11. Chen S.M., Chen H. H.: Fuzzy query processing in the distributed relational databases environment. *Database and Data Communication Network Systems*, Vol.1, Elsevier Science (USA) 2002.

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

Wpłynęło do Redakcji 19 lutego 2004 r.

Abstract

This article presents several possible forms of introduction to SQL queries notation of condition on degree of compatibility. In this work for both simple and nested SQL queries three forms notation of condition on degree of compatibility are discussed.

In the first chapter short introduction is presented.

The chapter two describes three forms of introduction to simple SQL queries notation of condition on degree of compatibility: joined, functioned and operated.

In the third chapter application of condition on degree of compatibility in simple SQL queries is presented. Every fuzzy condition is illustrated by appropriate example.

The fourth chapter presents usage condition on degree of compatibility in nested queries.

In the fifth chapter is a short summary.

Adres

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