

Bożena MAŁYSIAK  
Politechnika Śląska, Instytut Informatyki

## INTERPRETACJA ROZMYTYCH WARUNKÓW FILTRUJĄCYCH W ZAPYTANIACH SQL

**Streszczenie.** W artykule przedyskutowano proces interpretacji warunków filtrujących w zapytaniach SQL. Wyodrębniono podział na warunki koniunkcyjne, dysjunkcyjne oraz mieszane. Każdy taki przypadek został dokładnie przeanalizowany. Kilka z nich zostało zilustrowanych konkretnym przykładem.

**Słowa kluczowe:** pytania SQL, rozmyte warunki filtrujące, baza danych

## FILTERING CONDITIONS INTERPRETATION IN SQL QUERIES

**Summary.** This article presents interpretation process of filtering conditions in SQL queries. Filtering conditions were divided on conjunctions, disjunctions and mixed. Every of presented coincidence was analysed very precisely. Some of them was illustrated by example.

**Keywords:** SQL queries, fuzzy filtering conditions, database

### 1. Wprowadzenie

W klasycznych pytaniach zadawanych do bazy danych wiersz tablicy, do której odnosi się pytanie, spełnia lub nie kryteria podane w zapytaniu, a funkcja charakterystyczna (definiująca przynależność do zbioru) przyjmuje dwie wartości: 1 – gdy wiersz spełnia warunki pytania i należy do zbioru wynikowego odpowiedzi na pytanie i 0 w przeciwnym przypadku.

W pytaniach rozmytych stosujemy elementy teorii zbiorów rozmytych, a funkcją charakterystyczną określającą, w jakim stopniu wiersz w bazie danych spełnia kryteria wyszukiwania podane w pytaniu, jest funkcja przynależności, która może przyjmować wartości z przedziału  $<0, 1>$  [3, 4, 8].

Rozważmy proste zapytanie w języku SQL o postaci:

```
SELECT A1, . . . , Ak
FROM T
WHERE W;
```

gdzie  $W$  oznacza warunek filtrujący postaci:  $X$  **jest**  $A$  (gdzie  $X$  – kolumna tablicy bazy danych,  $A$  – zadana wartość).

Zauważmy, że proces wyszukiwania wierszy spełniających warunek zadania można potraktować jak proces wnioskowania: dla każdego wiersza tablicy  $T$ , na podstawie porównania wartości kolumny  $X$  zadaną wartością  $A$ , wnioskujemy o zaliczeniu wiersza do zbioru wynikowego lub nie.

W tym artykule rozważono podziały ze względu na rodzaj wartości występujących w warunku, jak również ze względu na spójniki łączące występujące w warunkach złożonych. Przypadki te dokładnie opisano w następnych punktach artykułu.

## 2. Podział warunków ze względu na rodzaj wartości $X$ i $A$

Ze względu na rodzaj wartości  $X$  i  $A$  warunki można podzielić na:

- 1) klasyczne, gdzie  $X$  i  $A$  przyjmują wartości ostre (dokładne),
- 2)  $X$  przyjmuje wartości ostre, a  $A$  wartości rozmyte – reprezentowane przez funkcję przynależności,
- 3)  $X$  przyjmuje wartości rozmyte, a  $A$  wartości ostre,
- 4)  $X$  i  $A$  przyjmują wartości rozmyte (reprezentowane przez dwa podzbiory rozmyte, dla których określone są funkcje przynależności).

Analiza każdego przypadku przebiega nieco inaczej, ale wynik ma zawsze taką samą ogólną postać: dla każdego wiersza (określającego wartość  $X$ ) należy wyznaczyć stopień zgodności wartości  $X$  z wartością  $A$  (lub ogólniej stopień zgodności, z jakim wartość  $X$  spełnia zadany warunek).

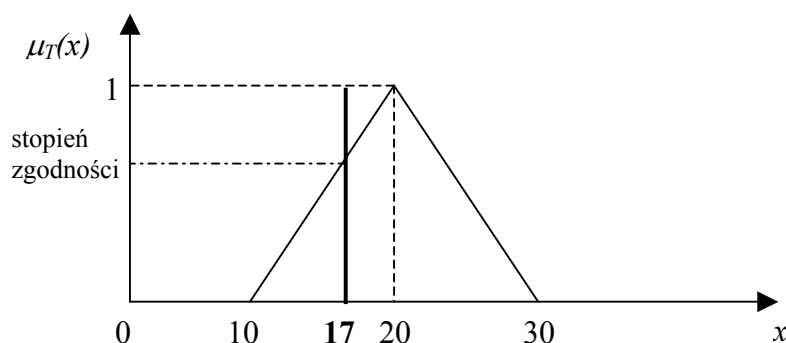
ad 1)

W przypadku kiedy wartości  $X$  oraz  $A$  są nierozmyte (warunek dokładny), wyznaczana jest wartość logiczna *True* (1) lub *False* (0) wyrażenia frazy *WHERE* i proces wnioskowania jest prosty: dla wartości *True* wiersz jest zaliczany do zbioru wynikowego, natomiast dla wartości *False* – nie jest.

ad 2 i 3)

W przypadku gdy jedna z wartości  $X$  lub  $A$  jest precyzyjna, a druga rozmyta, element rozmyty może być reprezentowany jako podzbiór rozmyty, dla którego zdefiniowana jest funkcja przynależności, a wartość dokładna reprezentowana jest przez linię pionową. Punkt

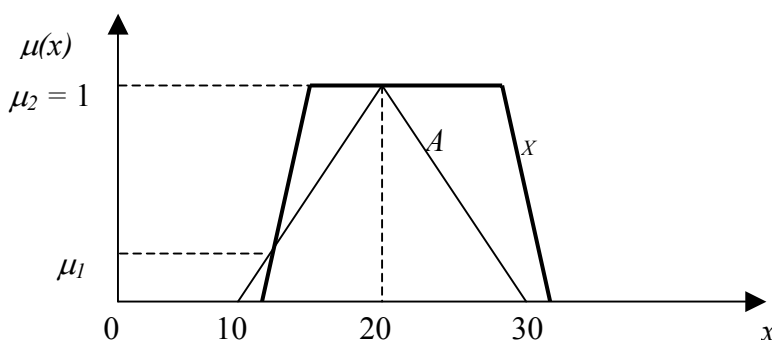
przecięcia funkcji przynależności i linii pionowej wyznacza stopień zgodności wiersza z kryteriami pytania (rys.1) [4].



Rys. 1. Przekięcie wartości rozmytej *około 20* z wartością precyzyjną 17  
Fig. 1. Intersection between fuzzy value *about 20* and crisp value 17

ad 4)

- a) W przypadku, gdy obie wartości  $X$  i  $A$  są rozmyte, (reprezentowane przez dwa podzbiory rozmyte, dla których określone są funkcje przynależności), wtedy stopień zgodności wiersza z kryteriami pytania wyznaczany jest jako przecięcie dwóch funkcji przynależności. Gdy występuje wiele punktów przecięcia brany jest pod uwagę ten o największej wartości [1]. Ilustruje to rys. 2. Pod uwagę brany jest stopień zgodności  $\mu_2$ , gdyż jego wartość jest większa niż  $\mu_1$ .



Rys. 2. Przekięcie wartości rozmytej  $A$  *około 20* zdefiniowanej przez trójkątną funkcję przynależności z wartością rozmytą  $X$  zdefiniowaną przez trapezoidalną funkcję przynależności [4]

Fig. 1. Intersection between fuzzy value  $A$  *about 20* defined by triangular membership function and fuzzy value  $X$  defined by trapezoidal membership function

- b) Jeżeli wartość kolumny  $X$  bieżącego wiersza wyrażona jest w postaci przedziału, a wartość  $A$  wyrażona za pomocą funkcji przynależności, to dla każdej wartości kolumny należy wyznaczyć stopień zgodności, a następnie wybrać ten o największej wartości [1]. Przypadek ten odnosi się do danych o typach przeliczalnych.

Wymienione przypadki dotyczą prostego warunku porównującego jedną wartość z drugą. Klauzula *WHERE* w klasycznej instrukcji *SQL* może jednak zawierać wiele warunków porównania połączonych poprzez operatory koniunkcji *AND* (*iloczyn logiczny*), dysjunkcji *OR* (*suma logiczna*). Kolumny mogą również być poprzedzone operatorem negacji *NOT* [5].

Szczegółową analizę wybranych przypadków przedstawimy, natomiast ogólnie można powiedzieć, że w takiej klauzuli *WHERE* wartości iloczynu lub sumy rozmytej mogą być obliczane na różne sposoby, a najprostszym jest zastosowanie *norm Zadeha* (*t-norma* dla operacji iloczynu polega na wyznaczeniu wartości minimalnej porównywanych stopni zgodności, *s-norma* dla operacji sumy polega na wyznaczeniu wartości maksymalnej). Można również zastosować inne operatory *s-normy* i *t-normy* [6, 7]. Natomiast gdy kolumna poprzedzona jest operatorem negacji *NOT*, wartość stopnia przynależności wyznaczana jest na przykład poprzez odjęcie od 1 początkowej wartości stopnia przynależności; można zastosować również inne metody.

### 3. Podział warunków ze względu na rodzaj spójników je łączących

Ze względu na rodzaj spójników łączących warunki filtrujące, warunki złożone można podzielić na:

- 1) warunki koniunkcyjne:
  - i.  $W_1 \text{ AND } W_2 \text{ AND } W_3 \dots W_N$ ,
- 2) warunki dysjunkcyjne:
  - i.  $W_1 \text{ OR } W_2 \text{ OR } W_3 \dots W_N$ ,
- 3) warunki mieszane

#### 3.1. Warunki koniunkcyjne

Dla warunków koniunkcyjnych o postaci:

$$W_1 \text{ AND } W_2 \text{ AND } W_3 \dots W_N,$$

można wyodrębnić następujące przypadki:

- **wszystkie  $W_i$  dają wartość *True* lub *False*** (warunki dokładne), wtedy spójnik *AND* pełni rolę iloczynu logicznego,
- **jeden z warunków  $W_i$  jest rozmyty** - proces filtracji przebiega w tym przypadku następująco: w pierwszym kroku filtracja przebiega względem warunków dokładnych. Jeśli analizowany wiersz nie spełnia któregoś z warunków dokładnych, to nie jest on brany pod uwagę i dalsza analiza nie jest prowadzona. Jeśli analizowany wiersz je spełnia, wyznaczany jest stopień zgodności wiersza

z warunkiem rozmytym  $W_i$ . Wyznaczona wartość stanowi całkowity stopień zgodności wiersza z warunkiem rozmytym.

- **dwa (lub więcej) warunki  $W_i$  są rozmyte**, wtedy, wykorzystując własność przemienności iloczynu, zgrupujemy warunki rozmyte razem, np.:

$$(W_k \text{ AND } W_m) \text{ AND } W_n \dots$$

Proces filtracji przebiega w tym przypadku następująco: w pierwszym kroku filtracja przebiega względem warunków dokładnych. Jeśli analizowany wiersz je spełnia, wyznaczany jest całkowity stopień zgodności tego wiersza z warunkami rozmytymi  $W_k$  i  $W_m$ .

Całkowity stopień zgodności dla tak połączonych warunków rozmytych wyznacza się poprzez obliczenie iloczynu rozmytego wyznaczonych stopni zgodności dla poszczególnych warunków rozmytych  $W_k$  i  $W_m$  (w tym celu można zastosować wybrany operator *t-normy*).

### 3.2. Warunki dysjunkcyjne

Dla warunków dysjunkcyjnych o postaci:

$$W_1 \text{ OR } W_2 \text{ OR } W_3 \dots W_N,$$

można wyodrębnić następujące przypadki:

- **wszystkie  $W_i$  są dokładne**, tzn. dają wartość *True* lub *False* (prawdy lub fałszu), wtedy spójnik *OR* pełni rolę sumy logicznej,
- **jeden z warunków  $W_i$  jest rozmyty** proces filtracji przebiega w tym przypadku następująco: w pierwszym kroku filtracja przebiega względem warunków dokładnych. Jeśli którykolwiek z nich jest spełniony, to całkowity stopień zgodności wiersza jest zawsze równy 1 i wiersz jest zaliczany do zbioru wynikowego. Jeśli analizowany wiersz nie spełnia żadnego z nich, wyznaczany jest stopień zgodności tego wiersza z warunkiem rozmytym.
- **dwa (lub więcej) warunki  $W_i$  są rozmyte**, wtedy, wykorzystując własność przemienności sumy, zgrupujemy warunki rozmyte razem, np.:

$$(W_k \text{ OR } W_m) \text{ OR } W_n \dots$$

Proces filtracji przebiega w tym przypadku następująco: w pierwszym kroku filtracja przebiega względem warunków dokładnych. W przypadku gdy ani jeden warunek dokładny nie jest spełniony, całkowity stopień zgodności wiersza jest równy całkowitemu stopniowi zgodności z warunkami rozmytymi. Całkowity stopień zgodności dla tak połączonych warunków rozmytych wyznacza się poprzez obliczenie sumy rozmytej stopni zgodności każdego z warunków (w tym celu można zastosować wybrany operator *s-normy*).

### 3.3. Warunki mieszane

Jest to przypadek najbardziej ogólny. Należy wtedy odrębnie pogrupować warunki dokładne i warunki rozmyte. Dla pierwszej grupy należy wyznaczyć końcową wartość logiczną (*True* lub *False*) warunku, dla drugiej natomiast należy wyznaczyć całkowity stopień zgodności z warunkami rozmytymi. Na końcu w zależności od rodzaju spójników łączących warunki rozmyte z warunkami dokładnymi należy wyznaczyć całkowity stopień zgodności wiersza z kryteriami pytania.

Wyznaczanie stopni zgodności dla warunków rozmytych w instrukcji *SELECT* przedstawia tabela 1 [2, 5]:

Tabela 1

Wyznaczanie stopni zgodności w instrukcji *SELECT*

fraza <i>WHERE</i>	<b>AND</b>	<i>t-norma</i>
	<b>OR</b>	<i>s-norma</i>
	<b>NOT</b>	<i>1-<math>\mu</math> lub inne</i>

Fragment gramatyki definiujący w sposób formalny zapis warunku rozmytego ma następującą postać:

```

...
<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
...

```

W pewnych sytuacjach wiersz może należeć do tabeli ze zdefiniowanym już wcześniej stopniem zgodności  $\mu$ , a ponadto można wyznaczyć wartość stopnia, z jakim wiersz spełnia warunki rozmyte  $\mu_l$  zadanego pytania. W tym przypadku całkowity stopień przynależności wiersza określony jest jako wartość  $\min \{\mu, \mu_l\}$ . Przypadek ten zostanie zilustrowany w przykładzie w punkcie 4.2

W wyniku wyznaczenia stopni zgodności dla frazy *WHERE* powstaje zbiór wierszy wynikowych. Zbiór ten może zawierać wszystkie wiersze tabeli (jeśli w pytaniu nie było ostrych warunków filtrujących nałożonych na ostre dane), a zawarte w nim wiersze mogą w różnym stopniu spełniać kryteria pytania. Może więc powstać potrzeba wybrania tylko części wierszy ze zbioru wynikowego w celu udostępnienia ich użytkownikowi.

Istnieją różne kryteria takiego wyboru [6], zwanego procesem wyostrzania. Najprostszy jest wybór wiersza o maksymalnym stopniu zgodności. Do innych należy na przykład wybór

wierszy o stopniu zgodności przekraczającym ustaloną wartość progową. Powyższe rozważania zostaną w następnym punkcie zilustrowane przykładami.

## 4. Przykłady ilustrujące rozważane przypadki

### 4.1. Baza danych zawiera rozmyte dane i do bazy danych zadawane jest również rozmyte pytanie

Rozważmy przypadek najbardziej ogólny, gdzie baza danych zawiera nieprecyzyjne (rozmyte) dane i do bazy danych zadawane jest również nieprecyzyjne (rozmyte) pytanie.

Założmy, że w systemie istnieje baza danych *ZAKŁADY*, a w niej tabela *Zapotrzebowanie* (tabela 2), zawierająca szacunkowe informacje o rocznych potrzebach zakładów w zakresie papieru, tonerów oraz płytek CD. W kolumnach *papier* oraz *toner* przechowywane są wartości rozmyte zdefiniowane za pomocą odpowiednich funkcji przynależności [9].

Przykładowe dane zapisane we fragmencie tabeli *Zapotrzebowanie* zawiera tabela 2.

Tabela 2

*Tabela Zapotrzebowanie*

Nr_zakl	toner	Papier
1	około 7	około 4
2	około 6	około 25
3	około 3	około 30
4	około 4	około 10
5	około 5	około 15

Niech kierowane do bazy danych pytanie ma postać: *Wyszukać zakłady, które złożyły zapotrzebowanie na dość dużo tonerów i jednocześnie niewiele papieru.*

W pytaniu tym obie wartości, zarówno ta określona w kryteriach zapytania, jak i wartość kolumny w tabeli są wartościami rozmytymi.

W języku SQL powyższe pytanie może być zapisane w następujący sposób:

```
SELECT nr_zakl
FROM zapotrzebowanie
WHERE toner jest dosc_duzo AND papier jest niewiele;
```

Założmy, że wartości rozmyte *dosc\_duzo tonerów* i *niewiele papieru*, występujące w warunkach filtrujących pytania, opisane są trapezowymi funkcjami przynależności.

Przyjmijmy, że funkcja przynależności dla podzbioru *dosc\_duzo tonerów* opisana jest wzorem:

$$\mu(x) = \begin{cases} 0 & \text{dla } x \leq 4 \\ \frac{x-4}{2} & \text{dla } 4 < x \leq 6 \\ 1 & \text{dla } x > 6 \end{cases}$$

Przyjmijmy, że funkcja przynależności do podzbioru *niewiele papieru* jest opisana wzorem:

$$\mu(x) = \begin{cases} 1 & \text{dla } x \leq 10 \\ \frac{20-x}{10} & \text{dla } 10 < x \leq 20 \\ 0 & \text{dla } x > 20 \end{cases}$$

Dla każdej z wartości kolumn rozmytych (*toner, papier*) również zdefiniowany jest zbiór rozmyty, reprezentowany przez odpowiednią funkcję przynależności. Dla danych z tabeli 2 dla wartości kolumny *toner* zdefiniowane są odpowiednio następujące trójkątne symetryczne funkcje przynależności:

$$- \mu(x)_{\text{okolo } 7} = \begin{cases} 0 & \text{dla } x \leq 5 \\ \frac{2-|x-7|}{2} & \text{dla } 5 < x \leq 9, \\ 0 & \text{dla } x > 9 \end{cases}$$

$$- \mu(x)_{\text{okolo } 6} = \begin{cases} 0 & \text{dla } x \leq 4 \\ \frac{2-|x-6|}{2} & \text{dla } 4 < x \leq 8, \\ 0 & \text{dla } x > 8 \end{cases}$$

$$- \mu(x)_{\text{okolo } 3} = \begin{cases} 0 & \text{dla } x \leq 1 \\ \frac{2-|x-3|}{2} & \text{dla } 1 < x \leq 5, \\ 0 & \text{dla } x > 5 \end{cases}$$

$$- \mu(x)_{\text{okolo } 4} = \begin{cases} 0 & \text{dla } x \leq 2 \\ \frac{2-|x-4|}{2} & \text{dla } 2 < x \leq 6, \\ 0 & \text{dla } x > 6 \end{cases}$$

$$- \mu(x)_{\text{okolo } 5} = \begin{cases} 0 & \text{dla } x \leq 3 \\ \frac{2-|x-5|}{2} & \text{dla } 3 < x \leq 7, \\ 0 & \text{dla } x > 7 \end{cases}$$

Dla danych z tabeli 2 dla wartości kolumny *papier* zdefiniowane są odpowiednio następujące trójkątne symetryczne funkcje przynależności:

$$- \mu(x)_{\text{okolo } 4} = \begin{cases} 0 & \text{dla } x \leq 2 \\ \frac{2-|x-4|}{2} & \text{dla } 2 < x \leq 6, \\ 0 & \text{dla } x > 6 \end{cases}$$



$$\begin{aligned}
 - \mu(x)_{\text{okolo } 25} &= \begin{cases} 0 & \text{dla } x \leq 22 \\ \frac{3-|x-25|}{3} & \text{dla } 22 < x \leq 28, \\ 0 & \text{dla } x > 28 \end{cases} \\
 - \mu(x)_{\text{okolo } 30} &= \begin{cases} 0 & \text{dla } x \leq 25 \\ \frac{5-|x-30|}{5} & \text{dla } 25 < x \leq 35, \\ 0 & \text{dla } x > 35 \end{cases} \\
 - \mu(x)_{\text{okolo } 10} &= \begin{cases} 0 & \text{dla } x \leq 7 \\ \frac{3-|x-10|}{3} & \text{dla } 7 < x \leq 13, \\ 0 & \text{dla } x > 13 \end{cases} \\
 - \mu(x)_{\text{okolo } 15} &= \begin{cases} 0 & \text{dla } x \leq 12 \\ \frac{3-|x-15|}{3} & \text{dla } 12 < x \leq 18. \\ 0 & \text{dla } x > 18 \end{cases}
 \end{aligned}$$

Stopień zgodności między kryterium pytania a wartością kolumny (reprezentowanymi przez dwa zbiory rozmyte, dla których określone są funkcje przynależności) wyznaczany jest jako przecięcie dwóch funkcji przynależności. Jeśli w wyniku wykonania tej operacji jest wiele punktów przecięcia, brany jest pod uwagę ten o największym stopniu zgodności [1, 7].

Na przykład w wyniku porównania wartości rozmytych *niewiele papieru* z liczbą rozmytą *okolo 15*, otrzymujemy dwa stopnie przynależności:  $2/7$  i  $8/13$ . Zgodnie z algorytmem, całkowity stopień przynależności wyznaczany jest jako wartość maksymalna spośród nich, tzn. wartość  $8/13$ .

Ponieważ w pytaniu są dwa warunki, proces ten jest powtarzany dla każdego z nich. Otrzymane wartości stopni zgodności przedstawione są w tabeli 3.

Tabela 3

Tabela *Zapotrzebowanie* z wyznaczonymi stopniami zgodności

Nr_zakl	toner	$\mu_{\text{toner}}$	papier	$\mu_{\text{papier}}$
1	okolo 7	1	Okolo 4	1
2	okolo 6	1	Okolo 25	0
3	okolo 3	$1/4$	Okolo 30	0
4	okolo 4	$1/2$	Okolo 10	1
5	okolo 5	$3/4$	Okolo 15	$8/13$

Następnie rozważa się spójnik, który występuje w klauzuli *WHERE* i odpowiednio wyznacza całkowity stopień zgodności wiersza z kryteriami zapytania. Po wykonaniu przedstawionych operacji zostaje wyznaczony całkowity stopień zgodności, z jakim każdy

wiersz spełnia warunki pytania. Wiersze z całkowitym stopniem zgodności, wyznaczonym jako wartość minimalna (dla spójnika *AND*), przedstawione są w tabeli 4.

Tabela 4  
Tabela *Zapotrzebowanie* z całkowitym stopniem zgodności

Nr_zakl	toner	$\tau$	Papier
1	około 7	1	około 4
2	około 6	0	około 25
3	około 3	0	około 30
4	około 4	$\frac{1}{2}$	około 10
5	około 5	8/13	około 15

W zależności od potrzeb użytkownika i zastosowania należy wyznaczyć najbardziej reprezentatywne wiersze tabeli, stosując odpowiednie metody wyboru wierszy.

#### 4.2. Baza danych dopuszcza istnienie kolumn, których zawartość tworzą stopnie przynależności, do bazy danych zadawane jest rozmyte pytanie

Przyjęty w artykule sposób interpretacji zapytań dopuszcza istnienie **kolumn, których zawartość tworzą stopnie zgodności** z pewnym kryterium. Rozpatrzmy na przykład bazę danych *ZAKŁADY*, która została rozbudowana poprzez utworzenie dodatkowej tabeli *Dobrzy pracownicy*. Tabela ta powstała na podstawie istniejącej już tabeli *Pracownicy* poprzez dodanie nowej kolumny *Dobry*. Wartości w tej kolumnie utworzono przez określenie dla każdego wiersza tej tabeli stopnia zgodności z kryterium *dobrzy pracownicy*. Przykładowe dane przedstawione zostały w tabeli 5.

Tabela 5  
Tabela *Dobrzy pracownicy*

Nr	Imie	Nazwisko	Wiek	Staz_pracy	Plec	Adres	Dobry
1	Jan	Kowalski	48	19	M	Zabrze	0,8
2	Kasia	Nowak	38	10	K	Chorzów	0,7
3	Marcin	Sowa	21	1	M	Gliwice	0,6
4	Jakub	Sroka	53	22	M	Kraków	0,3
5	Anna	Maj	47	8	K	Katowice	0,9

Do bazy danych zadawane jest następujące pytanie:

*Wyszukaj **dobrych** pracowników w wieku **około 50 lat** lub **dobrych** mających **staż pracy około 20 lat**.*

Dla wartości rozmytych *wiek około 50* i *staz\_pracy około 20* określono odpowiednie funkcje przynależności (wykorzystując funkcję Gaussa określoną w poprzednim rozdziale).

```
SELECT imie, nazwisko
FROM dobry_pracownicy
WHERE (wiek jest około 50 OR staz_pracy jest około 20) AND dobry;
```

Obliczone stopnie zgodności przedstawione są w tabeli 6.

Tabela 6

Tabela *Dobrzy pracownicy* z obliczonymi stopniami zgodności

Nr	Imie	Nazwisko	Wiek	$\mu_{50}(\text{Wiek})$	Staz_pracy	$\mu_{20}(\text{Staz\_pracy})$	Plec	Adres	Dobry
1	Jan	Kowalski	48	0,895	19	0,973	M	Zabrze	0,8
2	Kasia	Nowak	38	0,018	10	0,062	K	Chorzów	0,7
3	Marcin	Sowa	21	0,001	1	0,001	M	Gliwice	0,6
4	Jakub	Sroka	53	0,779	22	0,895	M	Kraków	0,3
5	Anna	Maj	47	0,779	8	0,018	K	Katowice	0,9

We frazie *WHERE* jako spójnik występuje operator *OR*, zastosujemy dla niego *s-normę* Zadeha. Stopień zgodności wiersza wyznaczony będzie jako wartość maksymalna spośród obliczonych cząstkowych stopni zgodności, ilustruje to tabela 7.

Tabela 7

Tabela *Dobrzy pracownicy* z obliczonymi stopniami zgodności względem pytania

Nr	Imie	Nazwisko	Wiek	$\mu$	Staz_pracy	Plec	Adres	Dobry
1	Jan	Kowalski	48	0,973	19	M	Zabrze	0,8
2	Kasia	Nowak	38	0,062	10	K	Chorzów	0,7
3	Marcin	Sowa	21	0,001	1	M	Gliwice	0,6
4	Jakub	Sroka	53	0,895	22	M	Kraków	0,3
5	Anna	Maj	47	0,779	8	K	Katowice	0,9

W kolejnym kroku określa się całkowity stopień zgodności wiersza poprzez wybranie wartości minimalnej spośród: obliczonego stopnia zgodności wiersza, a wprowadzonej wartości kryterialnej (stopnia zgodności z kryterium *dobrzy pracownicy*). Ponieważ między tymi stopniami zgodności stosuje się operator *AND*, zastosowano *t-normę* Zadeha. Całkowity stopień zgodności wiersza wyznaczony będzie jako wartość minimalna obu stopni zgodności. Wiersze z wyznaczonym całkowitym stopniem przynależności przedstawione są w tabeli 8.

Tabela 8

Tabela *Dobrzy pracownicy* z obliczonym całkowitym stopniem zgodności  $\tau$ 

Nr	Imie	Nazwisko	Wiek	$\tau$	Staz_pracy	Plec	Adres
1	Jan	Kowalski	48	0,8	19	M	Zabrze
2	Kasia	Nowak	38	0,062	10	K	Chorzów
3	Marcin	Sowa	21	0,001	1	M	Gliwice
4	Jakub	Sroka	53	0,3	22	M	Kraków
5	Anna	Maj	47	0,779	8	K	Katowice

## 5. Podsumowanie

W artykule przedyskutowano proces interpretacji warunków filtrujących zawierających wartości rozmyte. Proste warunki filtrujące podzielono ze względu na rodzaj występujących w nich wartości na klasyczne i rozmyte. Przy czym rozmytość może w warunku występować zarówno w wartościach kolumny, jak i w kryterium pytania.

Złożone warunki rozmyte podzielono na koniunkcyjne i dysjunkcyjne. Wszystkie przypadki zostały dokładnie przeanalizowane pod kątem interpretacji warunków filtrujących, a wybrane z nich zostały zilustrowane przykładami.

## 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. Małyśiak B.: Mechanizmy wnioskowania przybliżonego w bazach danych. Studia Informatica, Vol. 23, nr 4(51). Gliwice 2002.
4. Małyśiak B.: Wartości rozmyte w pytaniach SQL do baz danych. Studia Informatica, Vol. 24, nr 2A (53). Szczyrk 2003.
5. Małyśiak B.: Aproksymacyjne zapytania do baz danych. Studia Informatica, Vol. 23, nr 4 (51). Gliwice 2002.
6. Łachwa A.: Rozmyty świat zbiorów, liczb, relacji, faktów, reguł i decyzji. Akademicka Oficyna Wydawnicza Exit, Warszawa 2001.
7. Piegat A.: Modelowanie i sterowanie rozmyte. Akademicka Oficyna Wydawnicza Exit, Warszawa 1999.
8. Badurek J.: Logika rozmyta w bazach danych. Informatyka. Styczeń 1999
9. Małyśiak B.: Tworzenie typów, funkcji i operatorów rozmytych w SZBD Postgres. Konferencja Systemy Czasu Rzeczywistego. Krynica 2002.

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

Wpłynęło do Redakcji 19 lutego 2004 r.

**Abstract**

This article presents interpretation process of filtering conditions in SQL queries.

The first chapter is an introduction.

In the second chapter filtering conditions were divided on precise and imprecise. In general, three types of imprecision may arise:

- Imprecise Queries – precise data in database;
- Precise Queries – imprecise data in database;
- Imprecise Queries – imprecise data in database.

Interpretation process of intersection between fuzzy value and crisp value is illustrated on figure 1 and intersection between two fuzzy values first defined by triangular membership function and second by trapezoidal membership function is illustrated on figure 2.

In the next chapter filtering conditions were divided on conjunctions, disjunctions and mixed.

Some of them was illustrated by example in the fourth chapter. In subchapter 4.1 fuzzy query formulated to fuzzy database is analysed. In chapter 4.2 fuzzy query formulated to database which has additionally columns with membership degrees is analysed.

Every of presented coincidence was analysed very precisely.

**Adres**

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