

퍼지 데이터베이스 시스템을 위한 퍼지 질의어 연구(FSQL)* Fuzzy Structured Query Language for Fuzzy Database System

정은영, 신세영, 김승권, 유자영, 박순철
전북대학교 정보통신공학과 인터넷연구실
e-mail : moning@internet.chonbuk.ac.kr
Eun-Young Jung, Se-Young Shin, Seung-Gwon Kim,
Ja-Young Yoo, Soon C. Park
Internet Research Lab,
Dept. of Information and Communication Eng.,
Chonbuk National University

요약

우리가 일상적으로 사용하는 말 속에는 모호한 표현들이 많이 들어있다. 예를 들어, '젊다', '크다', '어느 정도' 등의 표현들은 정해진 값을 갖는 말들이 아니다. 가장 보편화된 RDBMS에서의 질의어인 SQL(Structured Query Language, 이하 SQL)은 데이터베이스에서 허용된 값, 즉 정량적인 값들에 대해서만 질의할 수 있도록 되어 있다. '젊은 여자' 혹은 '20세 정도의 여자'라는 질의는 할 수 없으며, '25세의 여자'라는 식으로 정확한 질의만이 허용된다. 그러나 정보량이 급증하고 있고, 정보가 곧 힘이 되는 지금, 일반 사용자들도 데이터베이스에서 자신이 원하는 정보를 얻어 낼 수 있어야만 하게 되었다. 따라서 본 논문에서는 일반 사용자들도 데이터베이스에서 일상적으로 사용하는 단어(이하 자연어)로 질의를 할 수 있도록 하는 FSQL에 대해 논의하고자 한다.

1. 서론

정보화 사회로 들어서면서 수많은 정보들의 가공 및 처리에 대한 문제가 중요시 되었고, 이에 대한 해결방안으로 데이터베이스가 등장하였다. 정보를 얼마나 더 많이 알고 있는가가 힘이 되는 지금, 일반 사용자들도 데이터베이스에서 자신이 원하는 정보를 찾아내는 것이 무척이나 중요하게 되었다. 그러나 데이터베이스에서 원하는 정보를 찾아내기 위해서는

질의 언어인 SQL이라든지, 데이터베이스 시스템 등에 대한 전문적인 지식이 필요하다.

우리들이 일반적으로 사용하는 언어는 '자연어'이다. 수 개 이상의 단어들과 조사와 접속사가 모여 한문장을 이룬다. 뿐만아니라 사용하는 단어들 중에는 정량적인 값을 갖는 경우도 있지만, '젊다', '크다' 등의 애매한 값을 갖는 경우도 있다. 따라서 본 논문에서는 정량적인 값을 갖는 질의 처리뿐만 아니라, 애매한 값을 갖는 질의도 처리할 수 있게 하는데 목표를 둔다. [1,2,3,4,5,6,7]

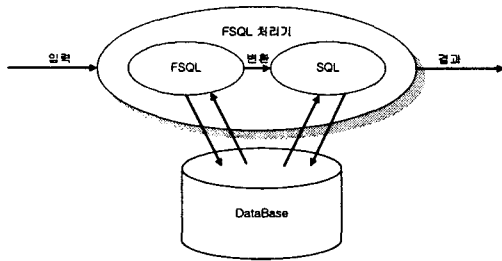
* 본 논문은 정보통신진흥연구원의 위탁으로 수행되고 있는 과제임(과제명: 인적사항 데이터베이스 조회를 위한 음성 인터페이스 퍼지 질의 처리)

본 논문은 다음과 같이 구성되었다. 2장에서는 본 논문에서 논하고자 하는 FSQL의 구조에 대해서 알아보도록 하며, 3장에서는 FSQL

의 정의 및 BNF형을 통해 FSQL에 대하여 구체적으로 알아보도록 한다. 4장에서는 이를 토대로 JavaCC로 구현한 parser에 대해 알아보고, 마지막으로 5장에서는 결과 및 향후 과제를 통해 앞으로 해결해야 할 문제점에 대해 논의하고 마치도록 하겠다.

2. 시스템의 구조

본 논문에서 논의 할 FSQL 시스템의 구조는 [그림 1]과 같다.



[그림 48] FSQL 시스템 구조

FSQL 형태로 입력된 질의는 데이터베이스 내의 Meta DB를 참조하여 SQL 형태로 변환되어 일반 SQL과 같은 형식으로 데이터베이스에서 결과값을 산출해 낸다. 본 시스템에서는 자연어 형태의 질의를 처리할 뿐만 아니라, 각각의 컬럼에 소속 척도 함수값을 두어 사용자에게 더 부합하는 문서와 그렇지 않은 문서를 차별화 할 수 있도록 하였다. [표 1]에서 보듯이 '나이가 젊은 사람'이라는 질의에 대하여 기존의 시스템은 결과값은 널 지 언정 누가 더 젊고, 덜 젊은지에 대해서는 차별화를 하지 않았다. 하지만 본 시스템에서는 소속 척도 함수값을 두어 'Jung'이라는 사람이 사용자가 요구한 질의에 더 부합함을 차별화함으로서 사용자의 편의를 높이도록 하였다.

[표 1] 소속 척도 함수값

기존의 경우		본 시스템의 경우		
Name	Value	Name	Value	Membership Value
Kim	27	Kim	27	0.84
Jung	22	Jung	22	0.99
Shin	15	Shin	15	0.80

이 밖에 다른 기능들은 3장에서 FSQL에 대해 논의하면서 설명하겠다.

3. FSQL

이 장에서는 FSQL 시스템에 앞서 모체라고 할 수 있는 SQL의 구조에 대해 간단히 알아보고 FSQL 시스템에 대해 논의하도록 하겠다.

3.1 SQL

SQL은 1970년대 초 System R 프로젝트의 일부분으로 구현되었던 Sequel 언어를 IBM이 SQL 언어로 재정의하면서 일반화 되었다.

SQL문의 기본 구조는 Select, From, Where의 세 개의 절로 이루어져 있다.

```
Select A1, A2, ..., An
From T1, T2, ..., Tm
Where P1 op P2 ... op Po
```

A_i : Attribute_name

T_i : Table_Name

P_k : Condition

op : Connection Operator(and, or, not)

[그림 2] 일반 SQL 형태

즉, [그림 2]는 '데이터베이스의 T₁, T₂, ..., T_m 테이블에서 P₁ op P₂ ... op P_o의 조건을 만족하는 사람(혹은 것)의 A₁, A₂, ..., A_n을 보여라'라는 것을 의미한다. 예를 들어 [그림 3]과 같은 SQL문은 'Background 테이블에서 나이가 25 이상이고, 키가 150인 사람의 이름과 주소를 보여라'라는 것을 의미한다.

```
Select Name, Address
From Background
Where age>=25 and height=150
```

[그림 3] 일반 SQL 형태의 예

위에서 알 수 있듯이 age, height는 Background 테이블의 Attribute로 되어 있으며, 25 혹은 150이라는 수치는 각 Attribute에 입력되어 있는 정보이다. RDBMS가 정량적인 값만 저장되도록 허용하므로, DBMS내의 값만 질의할 수 있는 SQL은 그 외의 질의를 할 수 없게 되는 것이다. 따라서 FSQL이라는 새로운 정의가 필요하다.

이 밖에도 데이터베이스에서 정보를 추가, 삭제, 갱신하는 여러 문장이 있으나, 본 논문의 FSQL 시스템은 정량적인 값만 저장하도록 되어있는 RDBMS를 활용하는 Vague형[7,8]이므로 애매한 값을 추가 혹은 갱신할 수는 없으며, 본 연구가 인적사항에 대한 검색으로 한정되어 있으므로 기본 구조에 대해서만 논의하도록 하겠다.

3.2 FSQL 정의

본 논문에서 제시하는 FSQL은 [그림 4]와 같은 형태로 이루어져 있다.

```
select A1, A2, ..., An bound K1
from T1, T2, ..., Tm
where [x1 θ F W1 y1 bound K2] op
      [x2 θ F W2 y2 bound K3] op ...
      [xo θ F Wo yo bound Ko]
```

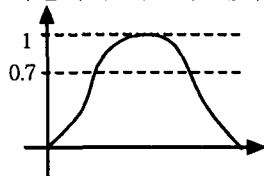
- A_i : Attribute_name
- T_j : Table_Name
- K_k : Bound Value
- x_a : Attribute_name for Searching
- y_b : Value
- W_c: Reserved Word(null or very, little, above, below)
- θ : Comparison Operator(<, ≤, =, ≠, >, ≥)
- F : ~ Operator
- op : Connection Operator(and, or, not)

[그림 4] FSQL 형태

SQL문과 달라진 점은 'bound'와 'F', 'W_c' 이다. 또, 같은 형태를 지니고 있으나 역할이 추가된 부분이 'op'이다. 이에 대해 자세히 살펴보자.

① bound

bound는 하한치의 역할을 한다.



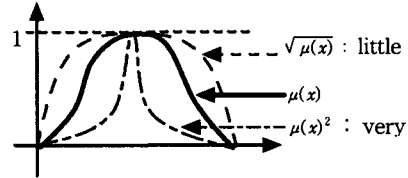
[그림 5] bound 0.7

bound는 [그림 5]에서와 같이 소속 척도 함수 혹은 결과값에 하한치를 둘 수 있게 하여 질의에 부합성이 적은 문서를 검색

대상에서 제외한다든지, 출력할 문서의 크기를 조절하는 역할을 한다. 즉, [그림 5]에서는 0.7이하에 해당하는 부분은 버리게 된다.

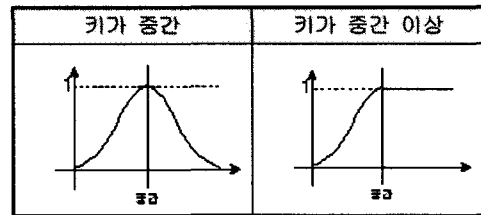
② 예약어(W_c)

질의시 '매우 젊은 여자를 찾아라'는 식으로 '젊음'의 정도를 밀하게 하고 싶거나, '조금 젊은 여자를 찾아라'는 식으로 '젊음'의 정도를 소하게 하고 싶을 경우가 있다. 이럴 경우에 'very' 혹은 'little'의 예약어를 사용하여 소속척도 함수의 정도를 바꾸어 줄 수 있다. 본 논문에서는 very의 경우 함수의 제곱을 취하고, little의 경우 함수의 제곱근을 취하였다. [그림 6]을 보면 이를 확인할 수 있다.



[그림 6] very, little 적용 예

또한 '이상의'의 의미를 갖는 'above'와 '이하의'의 의미를 갖는 'below'라는 예약어가 있다. 이는 소속 척도 함수를 일정 기준치에서부터 above의 경우는 오른쪽을, below의 경우에는 왼쪽을 '1'로 만들어 주는 기능을 한다.



[그림 7] above의 적용 예

③ ~ Operator(F)

이는 'fuzzy'함을 의미하며, fuzzy의 의미를 갖고자 하는 대상의 앞에 존재한다. 본 논문에서는 여타 다른 비교연산자는 퍼지할 수 없도록 한다. 즉, '거의 같다', '어느 정도 크거나 같다' 등의 연산자의 fuzzy 성질을 허용하지 않는다. 대신에 뒤에 나오는 조건이 fuzzy 하도록 한다. 따라서 '

키가 거의 180이다'라는 질의어는 'where height = ~180'으로 하여 '180'이라는 수치에 대한 소속 척도 함수값을 계산하도록 한다.

④ 결합연산자(op)

결합연산자 and, or는 일반 sql과 동일하나 결합시에 소속척도값의 계산이 요구된다. 어떤 사람이 나이가 젊다는 조건에 대해 0.5의 소속 척도값을 갖고, 키가 크다는 조건에 대해 0.9의 소속 척도값을 갖는다고 하자. '나이가 젊고 키가 큰 사람'이라는 질의와 '나이가 젊거나 키가 큰 사람'이라는 질의에 대해 이 사람은 상황에 따라 알맞은 하나의 소속 척도값을 갖아야 한다. 이에 대해 많은 연구가 활발히 진행되고 있고, 본 논문에서는 퍼지 연산에서 많이 발생하는 단일 피연산자 의존 문제와 부정적 보상 문제를 해결한 Fuzzy and, Fuzzy or 연산자[9,10,11,12]를 사용한다.

[표 2] Fuzzy and, Fuzzy or operation

Fuzzy and	$\gamma \text{MIN}(x, y) + \frac{(1-\gamma)(x+y)}{2}, 0 \leq \gamma \leq 1$
Fuzzy or	$\gamma \text{MAX}(x, y) + \frac{(1-\gamma)(x+y)}{2}, 0 \leq \gamma \leq 1$

①~④에서 논의한 바에 의해 [그림 8]과 같은 FSQ문은 'Background 테이블에서 나이가 25세 정도 되고, 키 150이상인 사람 중에서 소속 척도값 0.8이상인 사람만 출력하라'는 의미가 된다.

```
Select Name, Address bound 0.8
from Background
Where age=~25 and height = above 150
```

[그림 8] FSQ문 형태의 예

3.3 BNF

이제까지 논의된 FSQ문은 SQL형태를 그대로 수용하며 추가적으로 fuzzy에 관련된 몇가지 기능을 추가 하였다. 따라서 이에 대한 BNF형도 SQL의 BNF를 superset으로 하여 몇가지만 추가되었다. 다음은 SQL의 BNF에서 추가된 부분이다.

① 0과 1, 그리고 그 사이의 소수

<fnumeric literal> ::= 0 | 0[<period>[<unsigned integer>]] | 1

* <period>와 <unsigned integer>는 sql bnf에 정의되어 있음

② ~ 연산자의 추가

<FSQL special character> ::= <SQL special character> | <Tilde>
<Tilde> ::= ~

③ very 등의 예약어 추가

<Freserved word> ::= <reserved word> | bound | very | little | above | below

④ bound

<Fquery specification> ::= <query specification> | <query specification> <bound clause>

<Fwhere clause> ::= <where clause> | <where clause> <bound clause>

<bound clause> ::= bound <fnumeric literal>

4. FSQ처리 기 구현

FSQ 처리기는 크게 세 부분으로 구성된다. 먼저 FSQ를 분석하는 fsqparser와 현재의 상용 데이터베이스를 이용해 입력된 fsq 문장을 처리할 수 있도록 변환해주는 fsq2sql, 마지막으로, 결과 sql을 이용해 쿼리한 데이터를 유저에게 보여주는 fsqQuery application이다.

첫 번째로 'fsqparser'는 JavaCC를 사용해 구현하였다. JavaCC는 top-down 방식의 LL(k) 파서를 생성하는데, 정규식이나 문자열 같은 구문 명세와 문법 명세(BNF)를 한 파일에 함께 기술하기 때문에 읽기 쉬울뿐만 아니라 유지 보수가 쉽다는 이점이 있다. 또한 JavaCC는 순수 자바언어로 구현되었기 때문에 애플리케이션과 동일한 문법이 적용되고, 같은 입력 데이터를 동일하게 사용하므로 parsing time을 줄일 수 있다.

두 번째로 fsq2sql은, sql에서는 사용하지 않는 'bound'나 '~' 기호를 사용한 fsq 문장에 의한 쿼리가 가능하도록 fsq 문장을 sql 문장으로 변환하여 주는 기능을 한다. 이미 3절에서 정의된 metaDB를 참조하여 각각의 퍼지 성분의 단어에 대한 퍼지 정도의 값을 결정하고,

sql 에서는 사용이 불가능한 'bound', '~'를 적절한 sql 문장으로 변환한다.

[그림 9]는 나이에 대한 기록이 0세부터 145세까지 있고, metaDB에 속한 old의 0.8에 대한 value가 45세일 때를 가정하여 변환된 예이다.

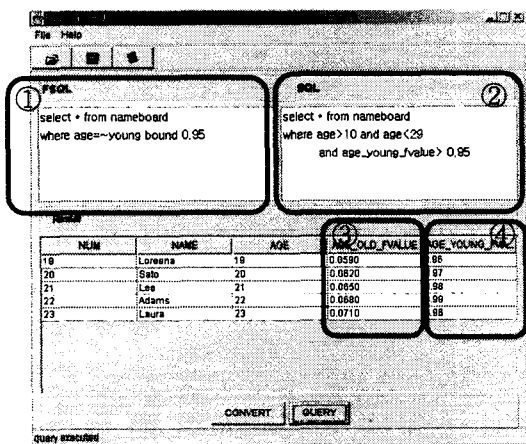
```

select manager
from employee
where age=~old bound 0,8
→
select manager
from employee
where 45<=age< 145

```

[그림 9] FSQL 처리기 변환 예

마지막으로 fsqQuery applicaton에서는 앞 단계의 결과인 sql 문장으로 데이터를 쿼리한다. [그림 10]은 fsqQuery applicaton을 실행시킨 모습이다.



[그림 10] FSQL 처리기

이제까지 정의한 FSQL 형태로 ①에 입력을 한 후, CONVERT 버튼을 누르면 ②번 창에는 Meta DB를 이용하여 FSQL을 SQL 형태로 변환한 문장이 나타나게 된다. 다시 QUERY 버튼을 누르면 아래의 창에 결과값을 낸다. 위의 경우는 num이 '1'인 사람은 age도 '1'로, num이 '20'인 사람은 age도 '20'으로 하여 넣은 100개의 데이터를 가지고, 나이가 젊은 사람에 대해 bound 0.95로 질의한 결과 값이다. ③,④는 각각 old와 young에 대한 소속 척도 함수 값을 나타낸다.

5. 결론 및 향후 연구

정보가 증가함에 따라 데이터베이스의 사용도 계속해서 증가할 것이며, 이를 얼마나 잘 활용하느냐의 문제 역시 계속 될 것이다. 사용자에게 얼마나 편리한 질의 환경을 제공할 것이냐의 문제가 남아 있는 한 자연어와 퍼지한 질의에 대한 처리 연구는 계속되어야 할 것이다.

본 연구는 주로 Meta DB를 이용하여 결과를 내다보니, 방대한 검색을 하는데 다소 무리가 있었다. 따라서 인적 사항이라는 한정된 사항에 대해서만 연구가 되었다. 이러한 부분이 개선되어야 할 것이며, 더불어 Meta DB와 소속 척도 함수 계산시에 소요되는 시간이 검색 효율을 낮추는 결과를 초래해서는 안 될 것이다. 더 나은 소속 척도 함수를 개발하는 연구도 계속 되어야 할 것이다. 또한 지식 기반 사회로 가면서, 이러한 정보들이 단지 데이터베이스에 저장되어 그대로 보여주는 역할에만 국한되는 것이 아니라, 의사 결정에 참여할 수 있는 시스템으로 발돋움하고 있다. 따라서 데이터 마이닝이나 퍼지 추론 기법을 도입하여 데이터베이스의 내용을 가공하여 추론한 결과를 낼 수 있는 시스템에 대한 연구도 계속되어야 할 것이다.

참고 문헌

- [1] 이광규, 전근환, 김흥기, "퍼지 질의어를 이용한 FDBL 모델", Journal of the Research Institute for Computer Science Vol. 4, No. 1, 1996. 2.
- [2] 이도현, 이광형, 김명호, "퍼지 질의를 위한 관계대수의 확장", 정보과학회논문지 제20권 제2호, 1993. 2.
- [3] 이석균, "신뢰도와 불확실성을 지원하는 확장 관계형 데이터 모델", 정보과학회논문지 제24권 제4호, 1997.4.
- [4] Buckles, B.P. and Petry, F.E., A Fuzzy Representation of Data for Relational Database, Fuzzy Sets and Systems 7, p213-226, 1982
- [5] Buckles, B.P. and Petry, F.E., "Information-Theoretical Characterization of Fuzzy Relational Databases", IEEE Trans, on System, M

an, and Cybernetics SMC-13, p74-77, 1983

[6] Motro, A. VAGUE, "A User Interface to Relational Databases that Permit Vague Queries", ACM Trans. on Office Information System em 6, p187-214, 1988

[7] S.C. Park, C.S. Kim and D.S, Kim, "Fuzzy Logic and its applications to engineering in formation sciences, and intelligent systems", KLUWER ACADEMIC PUBLISHERS, p407-415, 1995

[8] 이광형, 오길록, 퍼지이론 및 응용 II권, 흥릉과학출판사, 4장, 1991

[9] 고희대, "정보 검색 시스템에서 퍼지 집합 모델과 프로세서 구조에 관한 연구", 1997

[10] 정재근, "퍼지 질의어를 이용한 재고 시물레이션 시스템 구축에 관한 연구"

[11] <http://www.kordic.re.kr/~news/letter/15/n105.htm>

[12] http://borame.cs.pusan.ac.kr/jschoe/ir/ir_model.html

[13] Deyi li, Dongbo Liu, "A Fuzzy Prolog Database System",

[14] Zadeh, L. A., "Fuzzy Sets", *Information Control* 8, p.338-353, 1965

[15] 류근호, 이제환, "정보저장 및 검색", 2000

[16] 유윤희, "관계데이터베이스에 이용한 재고 시물레이션 시스템 구축에 관한 연구"

[17] 김병섭, "UML을 이용한 Fuzzy SQL 시스템의 설계 및 구현", 1999

[18] Klir G. J. and Folger, T. A., *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall International(UK) Limited, London, 1988.

[19] Sheno, S., Melton, A. and Fan, L. T., "An Equivalence Classes Model of Fuzzy Relational Database", *Fuzzy Sets and Systems* 3 8, p.153-170, 1990.

[20] Zemankova, M. and Kandel, A., "Implementing Imprecision in Information systems", *Information*



정은영

1995-1999 전북대학교 정보통신공학과
 1999-현재 전북대학교 대학원
 정보통신학과
 관심분야 : 데이터베이스, 네트워크 프로그래밍, 인터넷 프로그래밍



신세영

1995-1999 전북대학교 정보통신공학과
 1999-현재 전북대학교 대학원
 정보통신학과
 관심분야 : 데이터베이스, 네트워크 프로그래밍, Mobile Computing



김승권

1991-1999 전북대학교 정보통신공학과
 1999-현재 전북대학교 대학원
 정보통신학과
 관심분야 : Mobile Agents, 무선 인터넷 (WAP), Fuzzy -SQL/데이터베이스



유자영

1993-2000 전북대학교 정보통신공학과
 2000-현재 전북대학교 대학원
 정보통신학과
 관심분야 : Fuzzy Database, Mobile Agents



박순철

1972-1979 인하대 응용물리학과
 1986-1991 미국 루이지애나 주립대 전자계산 박사
 1991-1993 한국 전자통신 연구소 데이터베이스 연구실 과제 책임자
 1993-현재 전자정보공학부 부교수
 관심분야 : 데이터베이스, 네트워크 컴퓨팅, SAN(Storage Area Network),