HAVACILIK VE UZAY TEKNOLOJİLERİ DERGİSİ
OCAK 2011 CİLT 5 SAYI 1 (41-52)

BULANIK VERİTABANI VE BULANIK SORGULAR KULLANILARAK, İNSAN KAYNAKLARI ADAY SEÇME SİSTEMİ MODELİNİN OLUŞTURULMASI VE UYGULANMASI

Müh. Ütğm. Aydın ATA[*]

HHO Havacılık ve Uzay Teknolojileri Enstitüsü
34149, Yeşilyurt/İstanbul
aata@kkk.tsk.tr

Yrd.Doç.Dr.Hv.Müh.Alb. Sefer KURNAZ

HHO Havacılık ve Uzay Teknolojileri Enstitüsü
34149, Yeşilyurt/İstanbul
skurnaz@hho.edu.tr

Geliş Tarihi: *27 Temmuz 2010,* ***Kabul Tarihi:*** *24 Ocak 2011*

ÖZET

Son yıllarda, verilerin düzenli bir şekilde tutulması kadar, istenilen bilgiye doğru ve hızlı bir şekilde ulaşmak da önem kazanmaktadır. Veritabanı sistemlerinde verilere kesin sorgular üzerinden ulaşılabilir. Fakat insanların doğaları gereği sorgulamak istedikleri şey her zaman kesin değerler olmayabilir. Bu durum, yapılan sorgularda gereksiz birçok veri ile karşı karşıya kalınmasına sebep olmaktadır. Etrafımıza baktığımızda birçok şeyin kesin değerler üzerine kurulu olmadığını ve esneklik gerektirdiğini görüyoruz. Bazı bilgilerin kişiden kişiye değişen sübjektif değerler olması sorgulama işlemini daha da zorlaştırmaktadır. Bu nedenle veritabanı sorgulamalarında esneklik kazandıracak sorgu sistemlerine ihtiyaç duyulur. Bulanık sorgulamalar ve bulanık veritabanı, veritabanlarından istenen esnekliği kazandırmaktadır. Bu çalışmada, kesin değerler üzerinde çözülmesi zor bir konu olan insan kaynakları aday seçme sisteminin, bulanık veritabanı ve sorgular kullanılarak esnek bir yapıya dönüştürülmesi ve uygulanması gösterilmiştir. Bilgi tabanına girilen verilerin, oluşturulan kural tabanları vasıtasıyla değerlendirilmesi ve sonuçların istenen esneklikte raporlanması sağlanmıştır.

Anahtar Kelimeler: *Bulanık Veritabanı, Bulanık Sorgular, İlişkisel Veritabanı, Bulanık Kümeler, Veri Madenciliği.*

CREATING AND APPLYING A MODEL FOR HUMAN RESOURCES CANDIDATE SELECTION SYSTEM BY USING A FUZZY DATABASE AND FUZZY QUERIES

ABSTRACT

Retreieving the desired data accurately and quickly is as important as keeping it in an orderly manner. In database systems, data can be accessed by crisp queries. However, by nature, the user does not always use crisp queries. This leads to retrieval of redundant data for the queries. In nature, not everything is built upon exact values and thus needs flexibility. The subjective change of the data from one individual to another makes the query process harder. Thus, there is always a need for flexible query systems. Fuzzy queries and fuzzy databases give us the required flexibility. In this study, a human resources evaluation system was developed into a more flexible one by applying fuzzy databases and fuzzy queries. This is a hard subject to resolve by using crisp data. The data entered into the knowledge base was evaluated through the rule bases that were created and reporting with desired flexibity was achieved.

Keywords: *Fuzzy Database, Fuzzy Query, Relational Database, Fuzzy Sets, Data Mining.*

1. GİRİŞ

Gerçek dünya uygulamalarında bilgi çoğu zaman belirsiz ya da net değildir. Bu bilgilerin veritabanlarında saklanması gerekmektedir. Fakat sorun bu bilginin veri tabanında nasıl saklanacağı ve işleneceğidir. Bu sorunun cevabı, normal veritabanı sisteminin geliştirilerek belirsiz ve bulanık sorguları işleyebilecek hale getirmektir.

[*] Sorumlu Yazar

Bulanık veritabanları üzerine çalışmalar yaklaşık 20 yıldır devam etmektedir ve altı ana gruba ayrılmaktadır [1].

- Klasik(İlişkisel) veritabanları üzerinde bulanık sorgulama
- Bulanık veritabanları üzerinde bulanık sorgulama
- Bulanık veritabanlarının veri modellerinin oluşturulabilmesi için klasik veri modellerinin geliştirilmesi
- Bulanık kavramsal model uygulamaları
- Bulanık veri madenciliği teknikleri
- Bulanık veritabanı üzerinde uygulama geliştirme

Bulanık küme kuramını tanıtan Prof. Zadeh 'den beri bulanık yapılarda bu kuram kullanılmaktadır [2]. Veri tabanlarında bulanık sorgulamalar üzerinde ilk çalışma yapan kişi Tahani'dir. Tahani bulanık küme teorisini kullanarak veri tabanı sistemleri üzerinde bulanık sorgulamalar üzerinde çeşitli çalışmalar yapmıştır. Daha sonra Bosc [3], Wong ve Leung, Nakajima çeşitli fonksiyon ve komutlar yazarak SQL dilini daha da geliştirmişler ve daha esnek bir bulanık sorgu dili üzerinde çalışmışlardır. Fakat uygulama düzeyinde ilk ciddi çalışma Bosc ve Pivert tarafından yapılmıştır. Bu iki bilim adamı SQL dili üzerine yeni komutlar ve fonksiyonlar ekleyerek SQLF (SQL FUZZY) dilini yazmışlardır. Bu çalışma, bu konuda diğer çalışan akademisyenlerin önerdiği bulanık sorgulama sistemine paralellik göstermekte fakat daha gelişmiş yeni komut ve fonksiyonlar içermekteydi. Daha sonraki çalışmalarda SQLF dili daha da geliştirilmiştir [4-5]. Bulanık sorgulama komutları daha çok SQL dilinde SELECT komutuna yoğunlaşmakla beraber Galindo bu çalışmaları daha da ileri taşımıştır. Galindo, FSQL (FUZZY SQL) dilini geliştirmiştir [6]. Bu dil SQL 'in bir alt özelliği olan veri işleme dilinde (Data Manipulation Language) SELECT, INSERT, DELETE ve UPDATE komutlarının bulanıklaştırılmasını sağlamıştır. Ayrıca veri tanımlama dilinde (Data Definition Language) CREATE, ALTER ve DROP komutlarının bulanıklaştırılmasını sağlamıştır.
Bulanık sorgulamalar üzerine yapılan çalışmalarda α-eşik değeri yöntemi kullanılmaktadır. Bu yöntemde normal veritabanı yönetim sistemi üzerinde bir katman çalışması gerekmektedir. Eşik değeri yönteminde, bulanık operatör ve operandlar kullanılmaktadır. Normal veritabanı tablosundaki kayıt bir bulanıklaştırma sürecine girmekte ve bulanıklaştırılan değerler eşik değeriyle karşılaştırılmaktadır. İstenen koşulu sağlayan veriler seçilirken, koşulu sağlamayan değerler elenmektedir [7].

Veritabanı sistemlerinin esnekleştirilmesi konusunda çalışan birçok akademisyene göre veritabanı sistemlerinde bulanık küme teorisinin kullanılmasının araştırılması konusunda iki yol vardır.

Birinci yol, klasik veritabanı sistemlerine bulanık küme teorisi, olasılık teorisi ve bulanık mantık kullanılarak esnek bir sorgulama ara yüzünün geliştirilmesidir. (Bosc and Pivert, 1992, 1995; Bosc, Buckles, Petry, and Pivert, 1999; Dubois and Prade, 1997; Galindo, Medina, Cubero, and García,2000; Goncalves and Tineo, 2003, 2005; Kacprzyk, Zadrozny, and Ziolkowski, 1989; Ribeiro and Moreira, 1999; Tahani, 1977; Takahashi, 1991, 1995; Tineo, 2000)

İkinci yol, bulanık ve olasılık yöntemlerinin kullanılmasıyla oluşturulacak bulanık veritabanıdır. (Baldwin, Coyne, ve Martin, 1993; Bosc ve Pivert, 1997a, 1997b; Buckles ve Petry, 1985; Buckles, Petry, ve Sachar, 1986; Galindo, Medina, ve Aranda, 1999; Galindo, Medina, Pons, ve Cubero, 1998; Galindo vb.., 2006; Prade ve Testemale, 1984, 1987; Shenoi, Melton, ve Fan, 1990).

Yukarıdaki çalışmalar sonucunda baskın olan iki tür bulanık sorgulama aracı geliştirilmiştir. Bunlar;

FSQL (Galindo, 1999, 2007) ve SQLF (Bosc ve Pivert, 1995a) dir. Bu iki dil üzerine akademik çalışmalar yapılmaktadır [8].

FSQL dili, kesin sorguları, bulanık sorgulara dönüştürmek için ilişkisel veritabanlarında kullanılmak üzere tasarlanmıştır. Ayrıca bulanık verilerin veri modelinin tasarlanması için çalışmalar yapılmış ve Gelişmiş varlık ilişki modeli (Extended Entity- Relationship) Urrutia, 2003; Urrutia, Galindo, Jiménez, ve Piattini, 2006; Urrutia, Galindo, ve Piattini, 2002) geliştirilmiştir.

FSQL dilinin iki bilinen uygulaması mevcuttur. Bunlardan bir tanesi oracle üzerinde çalışan [9], diğeri PostgreSQL üzerinde çalışan uygulamalardır (Maraboli ve Abarzua, 2006).

FSQL dili açık kod bir dil olduğu için üzerinde çeşitli çalışmalar yapılmaya devam etmektedir. Fakat bu dilin en büyük dezavantajlarından biri, uygulama programları üzerindeki çalışmaların azlığıdır. Fakat 2008 yılında granada üniversitesinden R. A. Carrasco, F. Araque ,A. Salguero ,M. A. Vila turizm sektöründe kullanılmak üzere FSQL dili üzerinde bir bulanık uygulama geliştirmişlerdir [10]

Ayrıca bulanık diller üzerine yapılan geliştirme çalışmaları da devam etmektedir. Bu çalışmalar genellikle bu dilin yapısal durumu hakkındaki incelemelerdir [11].

Bu çalışmanın ikinci bölümünde bulanık küme teorisinin temeli için gerekli temel bilgiler verilmiştir. Üçüncü bölümde, çalışmanın temelini oluşturan kavramlar üzerinde durulmuştur. Dördüncü bölümde oluşturulan model ve uygulama ve bu uygulama

üzerinde yapılan deney çalışmaları gösterilmiştir. Son bölümde yapılan çalışma sonucu elde edilen sonuçlar ve öneriler bulunmaktadır.

2. BULANIK KÜME TEORİSİ

Küme, bir bütün olarak ele alınabilecek nesnelerden oluşmuş gruplardır. X[1] bir nesne kümesi olduğu varsayarsak, x bu X kümesini oluşturan elemanlardır. A kümesi ise X kümesinden üretilen altkümeleri temsil etmektedir. Bu durumda eğer x, A kümesinin elemanı ise (x,1) , değilse (x,0) ile gösterilir. Yani normal kümelerde eğer bir eleman o kümeye aitse 1 değilse 0 değerini almaktadır.[12]

Tanım: A kümesi X kümesinden üretilen bir altküme ve x, X kümesinin bir elemanı ise;

$$A = \{ (x,\mu_A(X)) \mid x \in X\} \qquad (1)$$

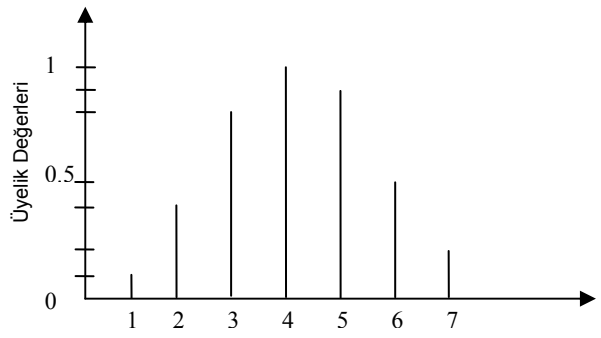
$\mu_A(X)$, A altkümesi için üyelik fonksiyonu olarak isimlendirilir. Bir üyelik fonksiyonu X kümesinde 0 ile 1 arasında değerler alabilir.

Örneğin X = {1,2,3,4,5,6,7} olduğunu varsayalım. Ailelerin kaç odalı evlerde mutlu oldukları ile ilgili A kümemiz aşağıdaki gibi ifade edilebilir.

A = { (1,0.1),(2,0.4),(3,0.8),(4,1),(5,0.9),(6,0.5),(7,0.2)}

A kümesi bize 1 odalı evlerde ailelerin 0.1 oranında yani oldukça az mutlu olduğunu, en çok ise 4 odalı evlerde mutlu olduklarını göstermektedir. Bu şekildeki kümelere ayrık kümeler adı verilir.

Bulanık kümelerin sayısı artınca bunları grafikler vasıtasıyla göstermek gerekir. A alt kümesinin grafiği şekil 1'dedir.



Şekil 1. Ayrık üyelik değer fonksiyonu.

Yukarıdaki şekilden görüldüğü gibi odaların sayısına göre memnuniyet üyelik değer fonksiyonu rahat bir şekilde görülmektedir. Fakat her zaman üyelik değer grafikleri ayrık olmaz. Örneğin genç insanlar kümesi ayrık olarak gösterilmez. Onun yerine kesintisiz bir grafik olması gerekir.

$$A = \begin{cases} \sum_{Xi \in X} \mu_A(x_i)/x_i & \text{(a)} \\ \int_X \mu_A(x)/x & \text{(b)} \end{cases}$$

(a) Eğer bütün nesneler ayrık küme elemanı ise
(b) Eğer bütün nesneler sürekli küme elemanı ise

Yukarıdaki gösterimdeki bulanık kümeleri gösteren toplam ve integral işaretleri burada gerçek anlamlarında kullanılmamakta, sadece toplam bulanık ifadelerin toplu bir gösterimini ifade etmektedir. [13]

Tanım: Bir A altkümesinin desteği, X kümesinin bütün x elemanları için

$$\text{Support (A)} = \{x|\mu_A(x) >0\} \qquad (2)$$

Tanım: Bir A altkümesinin merkezi, X kümesinin bütün x elemanları için

$$\text{Core (A)} = \{x|\mu_A(x) =1 \qquad (3)$$

Tanım: Bir A altkümesinin crossover noktası, X kümesinin bütün x elemanları için

$$\text{Crossover (A)} = \{x|\mu_A(x) =0.5\} \qquad (4)$$

Tanım: α-kesme değeri, bir bulanık kümede istenen sonucun alınması için gerekli eşik değerinin oluşturulmasını sağlar.

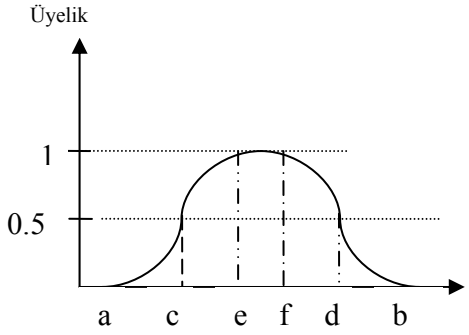
$$A_\alpha = \{x|\mu_A(x) \geq \alpha\} \qquad (5)$$

α-kesme değeri büyük eşit değil büyük olursa buna güçlü α-kesme adı verilir.

$$A'_\alpha = \{x|\mu_A(x) >\alpha\} \qquad (6)$$

Bu tanımlar şekil 2 üzerinde gösterilmiştir.

[1] Genellikle Tanım uzayı (Universe of discourse) olarak isimlendirilir.

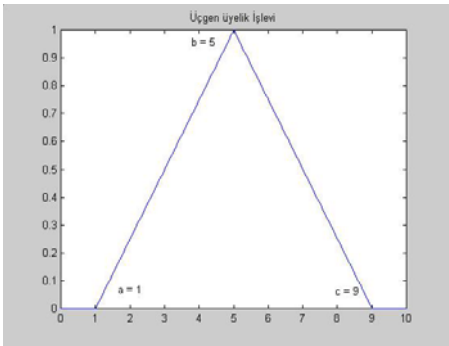


Şekil 2. [a,b] destek, [c,d] crossover, [e,f] merkez noktaları.

Bulanık kümelerdeki önemli konulardan bir tanesi elde edilen bulanık kümenin içindeki bir x elemanının üyelik değerinin bulunmasıdır. Tek boyutlu sistemlerde üyelik fonksiyonunun bulunmasında 4 çeşit yöntem sıklıkla kullanılır. Bunlardan 2 tanesi aşağıda tanıtılmıştır.

2.3.1 Üçgen Üyelik İşlevi

Üçgen üyelik işlevinde, üyeliği hesaplanmak istenen x değeri ve bu hesapta kullanılacak 3 parametre değeri kullanılmaktadır(Şekil 3). Örneğin a=1, b=5 c=9 olduğu aşağıdaki grafikte 6 sayısının üyelik fonksiyonu
μ(x) = (9–6)/9–5 = 3/4 = 0.75 dir. [14]



Şekil 3.Üçgen üyelik değer fonksiyonu.

$$üçgen(x,a,b,c) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & b \le x \le c \\ 0, & c \le x \end{cases}$$

Üçgen üyelik işlevi hesaplamaları işlemlerde kullanıldığı için bunun matematiksel formülü gereklidir.

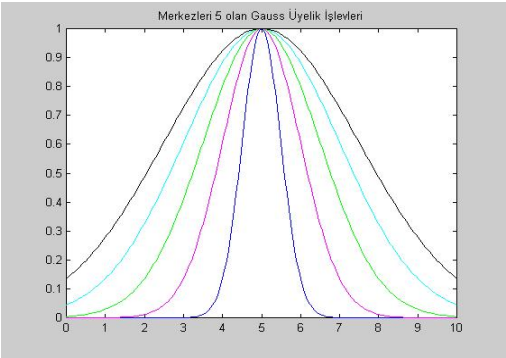
$$üçgen(x,a,b,c) = \max(\min(\frac{x-a}{b-a}, \frac{c-x}{c-b}), 0)$$

2.3.2 Gauss Üyelik İşlevi

Bir Gauss üyelik işlevi m(merkez) ve σ(varyans) gibi iki parametreyle tanımlanır: [14]

$$gauss(x,m,\sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}$$

Şekil 4'te değişik varyans değerlerine sahip ve merkezleri 5 olan üyelik fonksiyonları görülmektedir. (varyans sırasıyla 0.5, 1.0, 1.5, 2.0, 2.5 olarak alınmıştır.)



Şekil 4. Gauss Üyelik İşlevi.

3. BULANIK İLİŞKİSEL VERİTABANI MİMARİSİ

Bulanık ilişkisel veritabanının mimarisinin nasıl oluştuğunu ve bu mimariyi oluşturan temel elemanlar ve elemanlar arasındaki ilişkilerin anlaşılması önemlidir. Bu elemanlar farklı bir yapıya sahiptir. Veri yapıları(tablolar, viewler..), Veri tanımlama dili, veri işleme dili, veri vb. kavramlar bulunmaktadır.

Bu mimarinin görevi, bulanık, belirsiz veriyi depolamak ve işlemektir. Bunun yapılabilmesi içinde bir metot geliştirilmesine ihtiyaç vardır. Yani, bu yeni tip bilginin nasıl depolanacağına ve oluşturan bulanık veritabanının kullanıcı ile nasıl iletişim kuracağının belirlenmesine gerek vardır. Kullanıcı ile nasıl iletişim kurulacağı doğrudan veya dolaylı uygulamalar vasıtasıyla sağlanmalıdır. FSQL (Fuzzy SQL) dilinde bu yol SQL dilinin genişletilmesi ve düzenlenmesi vasıtasıyla sağlanmıştır. Zaten var olan ve yaygın olarak kullanılan SQL dilinin kullanılması hem öğrenmeyi kolaylaştırmakta, hem de yapısal duruma zarar vermemektedir. [15]

SQL dili bulanık ilişkisel veritabanlarında gerekli bulanık işlemleri yapabilmesi için düzenlenmiştir. Bu işlemler; bulanık özellik belirtme, uyumluluk derecesi belirleme, eşik değeri belirleme vb. kavramlardır.

FSQL dili yaygın olarak kabul edilebilmesi için istemci, sunucu mimarisinde çalışacak şekilde yazılmıştır. Bu dilin yaygınlaşması amacıyla dünyada yaygın olarak kullanılan bir ilişkisel veritabanı yönetim sistemini desteklemesi gerekmekteydi. FSQL, ORACLE veritabanı yönetim sistemini desteklemektedir. Fakat FSQL dilini destekleyen istemci yazılımları yaygın olarak oluşturulmadığı ve FSQL dilinin akademik ve gelişmeye açık bir dil olduğu için bu dil üzerinde uygulama geliştirmekte güçlüklerle karşılaşılmaktadır.

3.1. FSQL mimarisinin yapısı

Bulanık veritabanı ve bulanık sorguların nasıl çalıştığını anlayabilmek için FSQL mimari yapısının iyi anlaşılması gerekmektedir. Bu mimari, ilişkisel veritabanlarının bulanık veritabanlarına nasıl dönüştürüldüğünü açıklamaktadır.

3.1.1. İlişkisel veritabanı yönetim sistemi

Bulanık bir sorgu yapılmak istenirse veya FSQL içerisinde kullanılan bulanık yapılar kullanılırsa bu istekler SQL diline döndürülerek VTYS 'ye aktarılmaktadır. Diğer bir ifadeyle VTYS 'ye gönderilen istekler SQL veya FSQL dilinde gönderilebilir. Eğer FSQL dilinde gönderilmişse bu durumda ilgili FSQL ifade işlenmek üzere FSQL sunucuya gönderilmektedir [1].

3.1.2. Bulanık veritabanı

Bulanık veritabanı diğer veritabanları gibi bilgilerin ilişkisel olarak depolanmasını sağlar. Diğer veritabanlarından tek farkı oluşturulan tabloların bulanık verilerin depolanmasına izin vermesidir.

3.1.3. FMB (Fuzzy Metaknowledge Base)

İlişkisel veritabanlarında sözlük veya sistem katalog, veritabanı sistemi hakkında bilgileri tutar. Bir başka ifadeyle veri hakkında verilerdir. Burada kullanıcılar, izinler, veriye ulaşım bilgileri, kontrol vb. hakkında bilgi tutulmaktadır. Katalog bilgileri, klasik veritabanı ihtiyacını yerine getirmekte fakat bulanık değerler hakkında bir bilgi içermemektedir. Bu nedenle sistemin bulanıklaştırılabilmesi için veritabanı yönetim sisteminin katalog bilgilerinin de bulanıklaşmayı desteklemesi gereklidir. FSQL dilinde klasik katalog bilgilerini genişletmek ve bulanık değerleri desteklemek için FMB oluşturulmuştur.

3.1.4. FSQL sunucu

Bulanık sorgulamalarda, SQL dilinden farklı bir sorgulama dili olan FSQL dili kullanılmaktadır. Fakat veritabanı yönetim sistemimiz FSQL dilini anlamamakta, sorgu dili olarak SQL diline göre oluşturulmaktadır. Bu nedenle SQL dili ile veritabanı yönetim sisteminin birbirlerini anlamasını sağlayacak bir ara dönüştürücüye ihtiyaç vardır. FSQL sunucu, bu görevi yerine getirmektir. Bu görevi yaparken FMB içerisinde depolanmış bilgileri kullanır. FSQL sunucu, ORACLE veritabanı üzerinde çalışmaktadır. Bu nedenle, FSQL içerisinde çalışan fonksiyon ve yordamlar PL/SQL dilinde yazılmış ve oluşturulan bu modüller ORACLE sunucu üzerine depolanmıştır. Bu nedenle FSQL sunucu herhangi bir ORACLE platformunda çalışabilmektedir.

3.2. Oracle Veritabanı içerisinde bulanık bilgi gösterimi

Sistem belirsiz, bulanık değerleri destekleyen farklı tip özellik sınıflarına ayrılmıştır. Hangi sınıfın kullanılacağı, belirsiz bilginin durumuna bağlıdır. [16]

3.2.1. Bulanık özellik tipi 1

Bu özellik tipi bulanık veri tabanında kesin değerleri ifade etmek için kullanılır. Bu tipte FMB içerisinde etiket tanımlanabilir ve bu özelliği belirleyen bilgiler tanımlanabilir. Bu sayede klasik özellikler bulanıklaştırılabilir ve esnek sorgular üretilebilir. Fakat bu veri tiplerine bulanık değerlerin girilmesine izin verilmez. Çünkü bulanık tablolar yaratılırken bulanık özelliklerde yaratılır. Fakat bulanık özellik tipinde tablo yaratılırken özellik olarak kesin değerleri destekleyen özellik yaratılır.

Bu tip özelliğin normal veritabanı özelliğinden hiçbir farkı olmamasına rağmen, istenirse bulanık özellik kazandırılması çok büyük bir esneklik ve hız sağlar. Bu durumu sağlamak için bulanık etiketlerden faydalanılır. Bu tipin en sık kullanılabileceği alan veri madenciliğidir.

3.2.2. Bulanık özellik tipi 2

Bu özellik tipinde bulanık veri referans sıralı olarak alınır. Bu özellik tipinde uygun olarak daha önce açıklandığı gibi bulanık değerler girilir. Bu veri tipinde UNKNOWN, UNDEFINED ve NULL değerleri de girilebilir. Ayrıca, bu tipe kesin değerlerde girilebilir.

Tablo 1 bulanık tip2 özelliklerini göstermektedir. Bulanık tip 2 kısaca "F" ile ifade edilir ve 5 tane özelliği simgeler. Bunlardan FT tip kodunu belirlerken F1, F2, F3, F4 her verinin parametreleri depolar.

Tablo 1. Bulanık Özellik Tipi 2.

Değer Tipi	BULANIK ÖZELLİK TİPİ 2				
	FT	F1	F2	F3	F4
UNKNOWN	0	NULL	NULL	NULL	NULL
UNDEFINED	1	NULL	NULL	NULL	NULL
NULL	2	NULL	NULL	NULL	NULL
CRISP d	3	d	NULL	NULL	NULL
LABEL	4	FUZZY_ID	NULL	NULL	NULL
INTERVAL [n,m]	5	n	NULL	NULL	m
APPROXIMATE VALUE #d	6	d	d-marjin	d+marjin	marjin
TRAPEZOIDAL	7	α	β	γ	δ

Bu nedenle, Bulanık tip 2 özelliği bir veriyi bulanıklaştırmak için 5 adet klasik özellik tipinden faydalanmaktadır.

FT(Fuzzy Type), kayıda uygun veri tipini depolar. UNKNOWN (0) UNDEFINED (1), NULL (2), Crisp (3), LABEL (4), INTERVAL (5), APPROXIMATELY (6) ya da trapezoidal (7). Burada dikkat edilmesi gereken bir diğer nokta F ile gösterdiğimiz bulanık özelliğin arkasına T (tip) eklenmesidir. Sunucu özellikleri değerlendirirken özelliğin "T" ile bittiğini varsayar. Örneğin boy özelliğimizin tipini boyT olarak tanımlamamız gerekmektedir.

F1, F2, F3 ve F4 bulanık tipleri**,** bir bulanık özellik oluşturulurken bu özellik tiplerini ifade eder. Ana özelliğimizin sonuna 1, 2, 3 ve 4 sayıları eklenerek FT'nin parametrelerini oluştururlar.

UNKNOWN, UNDEFINED ve NULL, değerlerine parametre gerekmemekte bu nedenle parametre değerleri her zaman NULL olarak kalmaktadır. NULL ifadesinin bilinen klasik NULL ifadesidir. Bulanık bir ifade taşımamaktadır.

Kesin bir değer girildiğinde bu crisp bir özelliktir. Bu durumda bulanık tip1 ile aynı özelliğe sahiptir ve sadece bir kesin değer girilmesi gerekir. F1 özelliğine kesin değer girilir. F2, F3 ve F4 değerleri NULL olarak bırakılır.

F1 özelliğine, FMB içerisinde oluşturulacak etiketin numarası girilir. Bu numara değeri Label etiketi ile ifade edilir. Bu özellik tipi biraz daha hız kazandırmak için konulmuştur. Diğer özelliklere NULL değeri girilir.

Bu bulanık tipi kullanabilmek için 4 bulanık özelliğe de değer girmek gereklidir. Etiketleme işleminde de yamuk değerleri girilmekle beraber, etikete ihtiyaç olmadan bu şekilde de kullanılabilir.

Yukarıdaki tiplerden hangisinin seçileceği, veritabanı tasarımcısına bağlıdır. Bazı tipleri seçmek hızlı çalışmayı sağlarken, bazı tipler veritabanında daha az yer kaplamaktadır. Örneğin bir değer için sadece FT ve F1 fonksiyonları kullanılabilir fakat sistemin hızı biraz yavaşlar. Bunun sebebi, bulanık değerin bulunması için FMB içerisinde başka tablolara bakmak gerekmesidir. Fakat FT, F1, F2, F3 ve F4 kullanılarak yamuk fonksiyon değerleri direkt bulanık değere atanabilir. Bu durum sistemin daha hızlı çalışmasına fakat daha fazla veritabanı alanına sebep olacaktır.

3.3. Bulanık karşılaştırıcılar

SQL'de kullanılan klasik operatörlerin(<,>,= vb.) yanı sıra FSQL diline özgü operatörlerde mevcuttur. Bu operatörler tablo 2 'de gösterilmiştir.

Karşılaştırıcılar bir tablodaki bir özelliğin değeriyle bir sabiti karşılaştırabileceği gibi, aynı tipteki iki farklı özelliği de karşılaştırabilir. Possibility karşılaştırıcılar Necessity karşılaştırıcılardan daha genel sorgu sonuçları üretir. Diğer bir deyişle birinci karşılaştırıcılar daha çok sorgu sonucu üretirler.

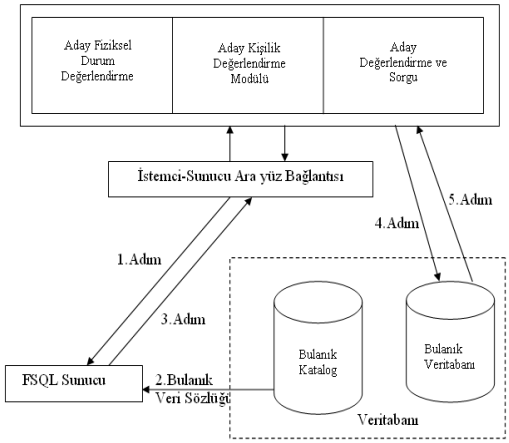
Tablo 2. Bulanık Karşılaştırıcılar.

Olasılık	Gereklilik	Anlamı
FEQ or F=	NFEQ or NF=	Olası/Gerekli Bulanık eşitlik
FDIF.F! or F<>	NFDIF.NF!= or NF<>	Olası/Gerekli Bulanık farklılık
FGT or F>	NFGT or NF>	Olası/Gerekli Bulanık büyüklük
FGEQ or F>=	NFGEQ or NF>=	Olası/Gerekli Bulanık büyük eşitlik
FLT or F<	NFLT or NF<	Olası/Gerekli Bulanık küçüklük
FLEQ or F<=	NFLEQ or NF<=	Olası/Gerekli Bulanık küçük eşitlik
MGT or F>>	NMGT or NF>>	Olası/Gerekli Bulanık çok büyüklük
MLT or F<<	NMLT or NF<<	Olası/Gerekli Bulanık çok küçüklük

4. İNSAN KAYNAKLARI ADAY SEÇME MODELİ VE UYGULAMASI

İnsan kaynaklarında aday seçme çok zahmetli ve zaman alıcı bir faaliyettir. Bunun nedeni, seçme adımlarının kesin değerlerle ifade edilmesindeki zorluktur. Seçme adımları bulanıktır ve bu işlemi kesin değerler üzerinde yapmak yanlış sonuçlara sebep olabilmektedir. Örneğin, bir öğrenci seçiminde bütün notları 90 üstünde olan bir öğrencinin bir dersten aldığı bir zayıf not o öğrenciyi klasik sistemde başarısız gösterebilir. Fakat genel olarak değerlendirmelerde o öğrenciyi başarısız değerlendirmemek gereklidir.

İnsan kaynakları aday seçme modeli, bulanık sorgu ve veritabanlarından faydalanarak, daha esnek sorgulamanın yapılmasını amaçlamaktadır. Bu sayede, daha hızlı, esnek ve ekonomik uygulamaların geliştirilmesi amaçlanmaktadır. Bazı uygulamalar kesin değerler içerisinde çalışmasına rağmen, bazı durumlar da kesin değerler içermeyen, muğlâk durumlar ile karşılaşılabilmektedir. Bu durumlarda bulanık sistemleri kullanmak daha avantajlıdır.



Şekil 5. Aday Seçme Modeli.

İnsan kaynakları aday seçme modeli, aday seçme sistemlerinde kullanılmak üzere tasarlanmış ve uygulanmıştır(Şekil 5). Amaç, klasik programlarla istenilen başarı sağlanamayan bir alanda bulanık uygulamalar yapmaktır.

Bu model, Aday seçme sistemi olarak geliştirilmiş ve ihtiyaçlara göre modüler tasarlanmıştır.

Model içerisindeki modüllerden birincisi adayların fiziki durumlarını değerlendirmek amacıyla tasarlanmıştır. Adayın boy, kilo ve boy-kilo arasındaki uyum kural tabanlarına göre belirlenmiştir.

İkinci modül, adayın kişilik bilgisini değerlendirmektir. Kişilik değerlendirme testleri yaygın olarak yapılmaktadır. Öncelikle adaylara 100 soruluk bir test sunulmaktadır. Bu test içerisinde adayın farklı kişilik yönlerini belirleyen 10 ayrı gruptan oluşmaktadır. Bu 100 soru, rast gele sorular halinde kullanıcıya sorulmakta, değerlendirme aşamasında, her kişilik yönü ayrı ayrı değerlendirilmektedir. Sonunda bu kişilik gruplarından istenen düzeyde olanları oluşturulan kural tabanına göre değerlendirilmekte ve istenen adayların hızla bulunması sağlanmaktadır.

Sorgu modülü, istenen adayların oluşturulması için kullanılmaktadır. Aday seçimlerinde bulanık olarak sorgu yapmaya imkân tanımaktadır.

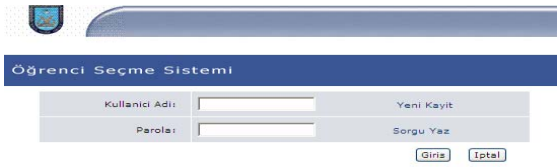
4.1. Aday seçme yazılımı

Aday seçme yazılımı, kesin değerlerle ifade edilmesinde sıkıntılarla karşılaşılan, bulanık değerleri hesaplamaya yönelik geliştirilmiş bir uygulamadır. Ülkemizde bulanık değerlendirme yapabilen uygulamaların olmaması, özellikle esneklik gerektiren uygulamalarda sıkıntılara sebep olmaktadır. Bu yazılım, bu eksikliği gidermektedir. Yazılım, fiziksel değerlendirme, kişilik değerlendirme ve sorgulama işlemlerini yapabilmektedir.

Adayların kişiliklerinin tespiti çok zor bir süreçtir ve bu konuda uzmanlar tarafından oluşturulmuş kişilik testlerine güvenilmektedir. Fakat bu testlerin yapılması ve değerlendirilmesi zahmetli bir iştir. Bu işin yapılabilmesi için bir kişilik testi hazırlama modülü oluşturulmuştur. Bu modül vasıtası ile testi hazırlayacak eğitmen veritabanında daha önce oluşturulmuş soruları alarak istediği sayıda test hazırlamakta, bu sorular istenilen zamanda web ortamı üzerinden adaylara uygulanmaktadır. Ayrıca, soruların rastgele olarak adaylara sunulması çok önemlidir. Her aday, soruyu farklı bir sırayla ekranda görmektedir. Ayrıca, test değerlendirmesi, klasik sınavlardan farklılık içermektedir. Kişilik testlerinde tek bir doğru cevap yoktur ve her cevabın belli bir değeri bulunmaktadır. Test sonunda, belli bir konuda sorulmuş soruların değerleri alınarak toplanmaktadır. Bu toplama işleminden sonra değerlendirme işlemi için iki ayrı yol izlenebilmektedir. Bunlardan birincisi, belli bir konuda alınan test değeri, daha önce oluşturulmuş kural tabanına göre değerlendirilebilir. İkinci yol ise, belli bir konuda alınan test sonucu, direkt olarak kural tabanında hesaplanmaz. Öncelikle bu konu, ilgili eğitmen tarafından bir norm işlemine tutulur. Bu işlem sonucunda elde edilen değer kural tabanıyla karşılaştırılır. Bu uygulama yazılımı her iki durumu da değerlendirmektedir.

Yazılıma ulaşabilmek için belli bir yetkilendirme yapılmıştır. Bunun sebebi, herhangi bir adayın test sorularına veya yetki gerektiren yerlere ulaşmasını engellemektir. Bu nedenle üç düzey tanımlanmıştır. Birinci düzey, yönetici aşamasıdır. Bu aşamada, kullanıcı program üzerinde her türlü yetkiye sahiptir. İkinci aşama, eğitmen aşamasıdır ve sadece yetkili bulunduğu sınıflara test hazırlayabilir ve testi başlatabilir. Ayrıca bu aşamada norm değer tabloları oluşturup, daha önce yapılan testlerin yeniden norm değerlerine göre hesaplanmasını sağlayabilir ve bulanık ve normal sorgulamalar yapabilir. Üçüncü yetki seviyesi adaylar içindir. Bu seviye de adaylar, kendileri için açılmış test var ise bunlara ulaşabilir. Aday, kendisi için açılmış bir testi en fazla bir defa alabilir ve test sonuçlandıktan sonra geri dönüş yapamaz.

Uygulamanın veritabanına bağlantısı otomatik olarak program tarafından yapılır. Yani, bütün kullanıcıların "huten" veritabanına bağlandığı varsayılmıştır.



Şekil 6. Aday Seçme Sistemi Girişi.

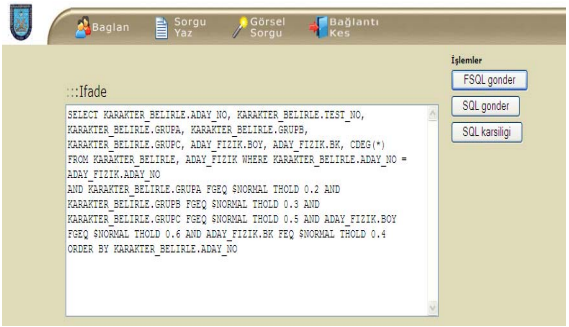
Web sayfasına bağlanıldığında kullanıcıya yetki seviyesi sorulmaktadır.(Şekil 6)

Yeni gelen adayların bilgisini girmek zahmetli bir iş olduğundan dolayı, bu işlemin adaylar tarafından yapılması sağlanmıştır. Yeni Kayıt seçeneğiyle adayların web programına kaydı sağlanır. Ayrıca sorgu yaz komutuyla yetkili personelin hızlı bir şekilde adaylar üzerinde sorgu yapması sağlanmaktadır.

Aday Seçme yazılımında adayların aldıkları notların norm değer tablosuna göre yeniden değerlendirilmesi gerekebilir (Şekil 7).

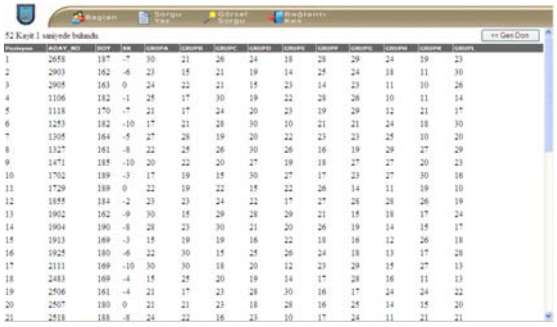
Şekil 7. Norm Değer Girişi.

Adaylar çeşitli testlerden ve başka seçici durumlarla test edildikten sonra istenen adayın istenen şartları sağlayıp sağlamadığının test edilmesi gerekmektedir. Bunun için şekil 8'deki sorgu modülü yazılmıştır.



Şekil 8. Bulanık sorgu çevrimi uygulama girişi.

Sorgu sonucu meydana gelen kayıtlar şekil 9'da gösterilmiştir.



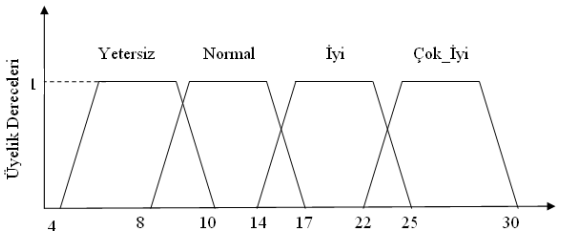
Şekil 9. Bulanık Sorgu Sonuçları.

4.2. Kural grafiklerinin elde edilmesi

Deney aşamasında en önemli durumlardan bir tanesi, kullanılacak kural grafiklerinin elde edilmesidir. Bu grafiklerin sistemi kullanan bir uzman tarafından oluşturulması sistemin performans ve güvenilirliğini arttıracaktır. Bu tez çalışmasında, kural tabanları İnsan Kaynakları uzmanı gözetiminde oluşturulmuştur. Kural grafikleri oluşturulurken yamuk fonksiyondan yararlanılmıştır.

Karakter belirlenmesi için 10 farklı kural grafiği, fiziksel durumun belirlenmesi için 3 farklı kural grafiği oluşturulmuştur. Karakter belirlenmesi için 10 farklı karakter özelliği değerlendirilmiştir. Bu karakter gruplarının hangileri olduğu belirtilmemiş sadece numara ile ifade edilmiştir. Aşağıda bir özelliğin karakter grafiği ve fiziksel durumun grafikleri gösterilmiştir.

Şekil 10'daki grafiğin bulanık kataloga aktarılması için öncelikle FCL(Fuzzy Control List) tablosuna bu özelliğin tanımlaması yapılmalıdır.



Şekil 10. Grup Test Değer Dağılımı.

INSERT into FCL values (t_NORM_TABLO,c_GRUPA,2,1,USER||'.NORM_TABLO.GRUPA');

Bu adımdan sonra, bu özelliği oluşturan dilsel karakterlerin atanması işlemi yapılmalıdır.

INSERT into FOL values (t_NORM_TABLO,c_GRUPA,0,'YETERSIZ',0);

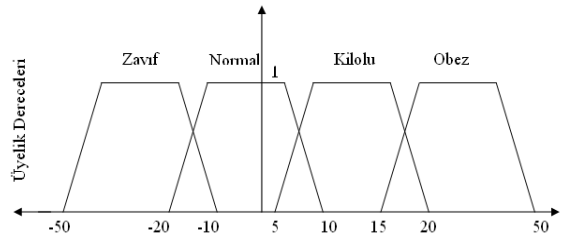
```
INSERT into FOL values
(t_NORM_TABLO,c_GRUPA,1,'NORMAL',0);

INSERT into FOL values
(t_NORM_TABLO,c_GRUPA,2,'IYI',0);

INSERT into FOL values
(t_NORM_TABLO,c_GRUPA,3,'COK_IYI',0);
```

Son olarak bu dilsel terimlere, bulanık değerlerin atanması gerekmektedir.

```
INSERT into FLD
values(t_NORM_TABLO,C_GRUPA,0,0,4,8,10);

INSERT into FLD
values(t_NORM_TABLO,C_GRUPA,1,8,10,14,17)

INSERT into FLD
values(t_NORM_TABLO,C_GRUPA,2,14,17,22,25)

INSERT into FLD
values(t_NORM_TABLO,C_GRUPA,3,22,25,29,30)
```

Ayrıca istenirse daha hassas ölçümler için gerekli dilsel etiketler çoğaltılabilir. Örneğin yukarıdaki grafik yapısı 4 dilsel karakter yerine daha fazla değerlerden oluşabilir. Yine, eğer istenirse dil niteleyicileri(az, biraz vs) değerler de atanabilir.

Kural grafikleri negatif değerlerde alabilir. Şekil 11'de boy-kilo arasında oluşturulan grafik negatif değerler almaktadır.



Şekil 11. Kilo - Boy Değer Dağılımı.

4.3. Yapılan sorgu deneyleri

Bu aşamaya kadar, kural tabanları oluşturuldu. Bu aşamadan sonra bulanık isteklerin girilmesi ve bu isteklerin durulanarak anlamlı sonuçlar üretilmesi aşamasına gelinmiştir. Bu aşamada, kural tabanları ile oluşturulan değerlere göre bulanık istekler girilecek ve sorgulamalar yapılacaktır.

Deney çalışmaları iki grupta yapılmıştır. Birinci grupta bulanık sorguların verimliliği test edilmiştir. Bu test aşamasında normal sorgulara göre bulanık sorguların kazandırdığı esneklik değerlendirmesi yapılmıştır. İkinci grup test çalışmasında, normal veritabanı sorgu hızları ile bulanık veritabanı sorgu hızları arasındaki farklar incelenmiş ve bulanık sorguların daha verimli çalışabilmesi için indeks üzerinden optimizasyon çalışması yapılmıştır.

Birinci grup testlerin sorgu deneylerinin yapılabilmesi için bilgi tabanlarının da oluşturulması gerekmektedir. Deney aşaması için 3335 aday kaydı girilmiştir. Bu adayların fiziksel bilgileri olan boy kilo ve boy-kilo arasındaki uyum da adaylar ile birlikte bilgi tabanına girilmiştir. Bu adayların test sonuçları gizli olduğu için, bu değerler yerine rast gele değeler üretilerek sorgulamalar yapılmış, bu sorgulamalar insan kaynakları değerlendirme uzmanı tarafından incelenmiştir.

Yapılan sorgularda, öncelikle karakter belirleme sorguları yapılmıştır. Daha sonra, fiziksel bilgilerin sorguları ve en son olarak da karakter ve fizik bilgileri birleştirilerek bulanık sorgulamaları yapılmıştır.

Karakter belirleme testlerinde 10 ayrı karakter belirlenmiştir. Öncelikle bu 10 ayrı karakteri ayrı ayrı sağlayan bir sorgu yazılmıştır. Alınan eşik değerleri, duruma göre değiştirilerek sistemin esnekliği artırılabilir.

Bütün bu sorgular sonucunda alınan değerler tablo 3'de özetlenmiştir.

Tablo 3. Deney Sonuç Çizelgesi.

Sorgu Adı	Kesin Sorgu Sayısı	Bulanık Sorgu Sayısı	Fark
Karakter Belirleme	640	761	121
Fiziksel Bilgi	265	1031	766
Hepsi	52	238	186

Bulanık veritabanı üzerindeki deney çalışmalarının ikincisi veritabanının çalışma hızı ve optimizasyon üzerinedir. Veritabanı dosyalarında verilerin hangi dosya organizasyonu içerisinde saklanacağının belirlenmesi sistem performansını etkileyecektir. Bu deney çalışmasında, dosya organizasyonu olarak heap dosya seçilmiş, performansı arttırmak için indeks yapısı kullanılmıştır. Deneyler üç grup olarak yapılmıştır. Birinci grupta, bulanık veri tipi 2 üzerinde indeks kullanılarak ve kullanılmadan yapılan deneylerdir. İkinci grupta, veri tipi 1 kullanılarak deneyler yapılmıştır. Son grupta, aynı koşullara yakın sorgular üreten normal veritabanı üzerinden sorgular gerçekleştirilecektir.

Bulanık sorgu performansı ikiye ayrılmaktadır. İlk olarak bulanık sorguların kesin sorgulara dönüştürülmesi işlemi için geçen zamandır. İkinci olarak alınan kesin sorgunun, tekrar ilişkisel veritabanına gönderilme işlemidir. Bu nedenle normal sorgulara göre bulanık sorguların daha yavaş çalışması normaldir. Fakat bu seviyenin kabul edilebilir bir seviyede olup olmadığı test edilmiştir.

Sorgu deneylerinden daha iyi sonuç alabilmek için bilgi tabanı bu deneylerde daha da genişletilmiştir. Bu sorgular için her bir tablo da 50.000 kayıt kullanılmıştır.

Yapılan deney sonuçları tablo'4 de özetlenmiştir.

Tablo 4. Deney Sonuç Çizelgesi.

Yapılan Deney	**Süre**
Bulanık Veri tipi(BV) 1	~2
BV 1 için Normal Veritabanı Sorgusu	~1
Bulanık Veri tipi 2	~6
Bulanık Veri tipi 2 indeks kullanımı	~4
BV2 için Normal Veritabanı Sorgusu	~2

5. SONUÇ VE ÖNERİLER

Bulanık veritabanı ve sorgulamalar, üzerinde akademik çalışmalar yapılan ve son zamanlar da önemi daha da artan konulardır. Fakat özellikle ülkemizde bu konu ile ilgili çalışmalar yok denecek kadar azdır. Bu nedenle, hem bu konudaki doküman sayısı, hem de akademik yayın miktarı sınırlıdır.

Bu çalışma ile bulanık veritabanı yapısı ve üzerinde geliştirilen bir uygulama gösterilmiştir. İlişkisel veritabanlarımızı bulanık veritabanlarına ve sorgu ortamına dönüştüren FSQL sunucu tüm yönleriyle incelenmiş ve bu sunucu kullanılarak bir aday seçme sistemi tasarlanmış ve sorgulama deneyleri yapılmıştır. Deney sonuçlarında, bulanık sistemlerin çok daha esnek sorgulamalar yapabildiği, bunun da özellikle esnek sorguların gerektiği yerlerde çok büyük fayda sağladığı görülmüştür. Ayrıca performans olarak yeterli düzeyde olduğu yapılan testler sonucunda anlaşılmıştır.

Geliştirilen "İnsan Kaynakları Aday Seçme Sistemi" isimli uygulama modeli ile aday seçme sistemlerinde bulanık işlemlerin daha esnek ve kolay bir şekilde yapılması sağlanmıştır. Adaylar tek bir noktadan değil birçok noktadan esnek bir şekilde sorgulanmakta ve bu sayede, en iyi adayların seçilmesi için gerekli ortam oluşturulmaktadır. Ayrıca geliştirilen programın ara yüzünde değişiklik yapılarak bulanıklık gerektiren her yerde bu uygulama istenen desteği sağlayacaktır.

Sistemin esnek çalışması ve performans durumunun ölçülmesi bu çalışmanın başka yerlerde kullanılıp kullanılmayacağının belirlenmesi açısından önem arz etmektedir. Esneklik çalışmasında, bir sorgulama sisteminde esnekliğin kazandırabileceği faydalar üzerinde durulmuştur. Bir seçme sistemini birden fazla duruma bakarak belirlenmektedir. Aday ne kadar fazla sayıda durum ile karşılaştırılırsa o kadar iyi bir adayın seçilmesi sağlanmış olmaktadır. Fakat bu durum beraberinde bazı zorluklar meydana getirmektedir. Bu zorluklardan bir tanesi, test edilen aşamaların her birinin ne kadar gerçeği yansıttığı durumudur. Seçilecek aday birden fazla seçim dinamiği içerisinde test edilirken bu testlerde bir anlık adaydan kaynaklanan (hastalık, heyecan vb.) düşüşler yaşanabilmektedir. Seçim aşamalarında bu düşüşlerin tespit edilmesi iyi adayın seçilmesi için çok önemlidir. Bu durumun kesin sorgularla elde edilmesi mümkün olmamaktadır. Belli bir esneklik içerisinde seçim adaylarının istenen koşulları hangi düzeyde sağladığı incelenmesi gerekmektedir. Bu durumu çok hızlı bir şekilde sağlayan bir uygulama programı sistem performansını ve istenen adayın seçilmesi olasılığını artıracaktır.

Test aşamasında 3335 adayın durumları incelenmiş ve esnek sorgulama vasıtasıyla karakter testlerinde normalde seçilen adayların % 19 oranında daha arttırılarak sistem içerisinde tekrar değerlendirilmesi sağlanmıştır. Bu adayların tekrar fiziksel değerlendirmeye tabi tutulması ve bu adaylardan üstün fiziksel duruma sahip olanların tekrar değerlendirilmesi sağlanmıştır. Bütün değerlendirme sonucunda kabul edilebilir düzeyde yapılan esnek sorgularda 186 adayın sistem tarafından seçilmesi sağlanmıştır. Bu adayların her birinin sorgusu yapılan konulara uyumlulukları incelenmiş ve sisteme tam olarak uydukları, ve seçilmeyi hak ettikleri görülmüştür.

Uygulaması yapılan sistemin istenen performansta çalışmasının kontrol edilmesi için performans deneyleri yapılmıştır. Yapılan bu çalışmanın, çok yüksek düzeyde veri ile çalışmadığı orta yoğunluklu bir ortamda çalışacağı varsayılmıştır. Bu nedenle elli bin ile yüz bin arasındaki kayıtlar üzerinde sistemin çalışma ve sorgu performansları incelenmiştir.

İncelemesi yapılan ilk çalışma bulanık veri tipi 2 üzerinde yapılan çalışmadır. Bu veri tipi normal veri tipine göre hafızada dört kat daha fazla yer kaplamakta ve bu nedenle sistemin performansını yavaşlatmaktadır. Bunun sebebi, bir veritabanı sorgulamasını en çok yavaşlatan unsurun disk üzerine ulaşım olmasından kaynaklanmaktadır. Veriler "page" içerisinde tutulmakta ve istenen sorgular yapılmak istendiğinde diskten bu alanların okunması ve gerekirse güncellenmesi gerekmektedir. Bulanık veri tipi 2, verilerin disk üzerinde daha fazla yer kaplamasından dolayı, disk ulaşım sayısı artmakta ve performans düşmektedir. Yapılan deneylerde elli bin kayda bulanık veri tipi 2 üzerinde yaklaşık altı

saniyede ulaşılmaktadır. Fakat aynı sonucu getirebilecek normal bir veritabanında bu sorgu yaklaşık iki saniye sürdüğü yapılan deneyler sonucunda görülmüştür. Bu veri tipinde sistem performansını artırmak için indeks optimizasyonu yapılmıştır. Öncelikle bulanık veri tipi 2 üzerinde daha çok bulanık tip(FT) ve "F1" alanları kullanıldığı için bu alanlara indeks kullanılmıştır. Bu durumda sistem performansı sadece bu alanların kullanıldığı durumlarda %50 düzeyinde arttırılmıştır. Ayrıca en çok sorgusu yapılan bulanık özellik kısımlarına indeks kullanılarak sistemin performansı aynı şekilde arttırılmaktadır.

Diğer veri tipi özelliklerinin sistem performansı incelenmiştir. Yapılan deneyler sonucunda bulanık tip 1'in performans üzerinde fazla bir kayba sebep olmadığı görülmüştür ve eğer veriler bulanık değil de bulanık tip 1 üzerinde çalışma yapılmasının bulanık tip 2 ye göre performansı %300 arttırdığı görülmüştür.

Uygulama aşamasında kullanılan FSQL sunucu, özellikleri bakımından daha da geliştirilebilecek bir durumdadır. Fakat şu haliyle bile uygulamaların ihtiyaçlarını rahatlıkla karşılamaktadır. Bu sunucunu yeni olması ve uygulama alanlarındaki eksikliği en büyük dezavantajdır.

Bulanık veritabanı ve sorgulamalar, veri madenciliği konusunda akademik çalışma yapacak akademisyenler için çok değerlidir. Bunu sebebi, bir çok veri madenciliği konusu yapısı gereği bulanık olmasıdır. Ayrıca FSQL dilinin veri madenciliğini desteklemesi ve FSQL kullanılarak veri madenciliği çalışmaları son zamanlarda yapılmaktadır.

FSQL sunucu yazılımının en büyük dezavantajı içerisinde öğrenen algoritmalar bulundurmamasıdır. Fakat bulanık sistemler için arka planda öğrenen algoritmaların çalışması önemlidir. FSQL sunucu içerisine böyle bir modül yazılması sistemin çok daha performanslı çalışmasını sağlayacaktır.

Bulanık veritabanı optimizasyonu hakkında gelecekte daha fazla çalışma olacağı değerlendirilmektedir. İleriki dönemlerde bu çalışmanın başka alanlara da uygulanması planlanmaktadır. Ayrıca bulanık veritabanı diline ek modüller yazılarak daha esnek hale getirilmesi veya bir bulanık sunucunun yazılması düşünülmektedir.

6.KAYNAKLAR

[1] Galindo, J. 2008. Introduction .and Trends to Fuzzy Logic and Fuzzy.Databases, in: Handbook of Research on Fuzzy Information Processing in Databases, Information Science Reference, p. 1-33, Hershey, PA, USA , 2008, vol. 1

[2] Terano, T. Sugeno.M. 1992. Fuzzy Systems Theory and Its Applications,Academic Press, Boston.

[3] Bosc, P. Galibourg, M. Hamon G. 1988. Fuzzy querying with SQL:extensions and implementation aspects, Fuzzy Set. Syst. 28 333–349.

[4] Yazici,A.Koyuncu, M. Fuzzy object-oriented database modeling coupled with fuzzy logic, Fuzzy Set. Syst. 89 (1) (1997) 1–26.

[5] Goncalves, M. Tineo L.2001. SQLf3: an extension of SQLf with SQL3 features, Proceedings of the 2001 IEEE International Conference on Fuzzy Systems 477–480.

[6] Carrasco, R., Vila, M.A., Galindo, J. 2003. FSQL: A flexible query language for data mining. Enterprise Information Systems, IV, 68-74. Hingham, MA: Kluwer Academic Publishers.

[7] Ma, Z.M., Yan L.,2007, Generalization of strategies for fuzzy query translation in classical relational databases, Science Direct, Information and Software Technology 49 172–180.

[8] Urrutia, A, Tineo, L, Gonzalez,C.,2008. FSQL and SQLf Towards a Standard in Fuzzy Databases , in: Handbook of Research on Fuzzy Information Processing in Databases, Information Science Reference, p.270-298, Hershey, PA, USA: , 2008, vol. 1.

[9] Galindo, J. 2007. FSQL (fuzzy SQL): A fuzzy query language. http://www.lcc.uma.es/~ppgg/FSQL Son Erişim:22 Ocak 2010.

[10] Carrasco, R. A., Araque, F., Salguero A., Vila, M. A.2008. Applying Fuzzy Data Mining to Tourism Area, in: Handbook of Research on Fuzzy Information Processing in Databases, Information Science Reference, p.563-585, Hershey, PA, USA: , 2008, vol. 1.

[11] Bouaziz, R. Chakhar S., Mousseau V., Ram S., Telmoudi A.2007. Database design and querying within the fuzzy semantic model, Science Direct, Information Sciences 177 (2007) 4598–4620.

[12] J.-S.R.JANG, C.-T.SUN, E.MIZUTANI, "Neuro Fuzzy and Soft Computing",Prince Hall",1-52,1997.

[13] Timothy J.Ross, "Fuzzy Logic with Engineering Applications Second Edition", John Wiley & Sons Ltd", 1-114, 2004.

[14] Okyay KAYNAK, HUTEN Bulanık Mantık Ders Notları.

[15] Galindo,J., Urrutia,A. Piattini, M. 2005. "Fuzzy Databases:Modeling, Design and Implementation", Idea Group Publishing, 341 sf., USA.

[16] Hassine M., Touzi, A. Galindo J., Ounelli ,H.2008. How to Achieve Fuzzy Relational Databases Managing Fuzzy Data and Metadata in: Handbook of Research on Fuzzy Information Processing in

Databases, Information Science Reference, p.351-380, Hershey, PA, USA: , 2008, vol. 1.

[17] Bordogna, G, Psaila, G. 2008. Customizable.Flexible.Querying.for.Classical.Relation al. Databases. , in: Handbook of Research on Fuzzy Information Processing in Databases, Information Science Reference, p.191-217, Hershey, PA, USA, 2008, vol. 1.

7. ÖZGEÇMİŞLER

Müh.Ütğm. Aydın ATA

2001 yılında Selçuk Üniversitesi Bilgisayar Mühendisliği Bölümünden mezun oldu. Kara Harp Okulundaki eğitiminden sonra 2002–2008 yılları arasında Mühendis Subay olarak Muhabere Elektronik ve Bilgi Sistemleri Okulunda görev yaptı. 2008 Yılında Kara Kuvvetleri Komutanlığı yurtiçi yüksek lisans programına seçilerek Havacılık ve Uzay Teknolojileri Enstitüsü(HUTEN) Bilgisayar Mühendisliği bölümünde yüksek lisans eğitimine başladı ve 2010 yılında mezun oldu. Halen K.K.K.'lığında bilgi sistem subayı olarak görev yapmaktadır.

Yrd.Doç.Dr.Hv.Müh.Alb. Sefer KURNAZ

1978 yılında Hava Harp Okulu Elektronik Mühendisliği bölümünden lisans, Ege Üniversitesi Bilgisayar Mühendisliğinden yüksek lisans, İstanbul Üniversitesi Bilgisayar Mühendisliğinden doktora derecesi aldı. Halen Hava Harp Okulu Komutanlığı Havacılık ve Uzay Teknolojileri Enstitüsü Müdürü olarak görev yapmaktadır.