Bulanık Mantık Temelli Esnek Sorgulama Aracı

S. İlhan, N. Duru
Kocaeli Üniversitesi Bilgisayar Müh. Bölümü, Kocaeli
silhan@kou.edu.tr, nduru@kou.edu.tr

Özetçe

Bu çalışmada, klasik veri tabanlarında bulanık sorgulama işlemini gerçekleştiren bir "esnek sorgulama aracı" geliştirilmiştir. Esnek sorgulama işlemi, veri tabanındaki kayıtlı tablolar üzerindeki nitelik değerlerinin bulanıklaştırılması işlemi ile gerçekleştirilir. Geliştirilen sorgulama aracı, burs işlemlerinin yapılabilmesi için kayıtların tutulduğu örnek bir "Burs" veri tabanı üzerinde uygulanmıştır Bu işlem sonucunda, bulanık sorgu yönteminin daha esnek bir yaklaşım sunduğu ve elde edilen sonuçların daha tahmin edilebilir olduğu görülmüştür.

Absract

In this paper, a software tool for enabling fuzzy query from a classical database is introduced. By using this tool, some attributes of a database table can be fuzzified and supplementary database, which includes fuzzy values, is formed. Developed software tool is applied in a sample database is including some fields about the bursary(grant) informations. It is concluded that, the fuzzy query method is more flexible and results of such query are more predictive.

1. Giriş

İnsan düşüncelerinin büyük çoğunluğu bulanıktır. İnsan, çevresinde algıladıklarını tanımlarken ya da bunları beyninde sınıflandırırken kesin ifadelerin yanında bulanık ifadelere de başvurur. Hissedilen bir sıcaklığı yalnızca sıcak ve soğuk diye ayırmaktan çok, ılık, çok sıcak, çok soğuk gibi ifadeler ile belirtiriz. Günlük yaşantımızda algılarımızı kesin ifadeler ile anlatmaktan çok bulanık ifadelere başvurduğumuz açıktır. Bulanık ifadelere başvurmanın bir diğer nedeni de, zaman içerisinde veriler aynı kalsa dahi, insanın yaptığı değerlendirmelerde değişikliklerin olabileceğidir. Ya da veriler, insandan insana, sistemden sisteme farklılık gösterebilirler. Örneğin, bazı insanlar tarafından aylık geliri bir milyar olan zengin olarak kabul edilirken başka bir grup insan için aylık geliri beş milyar ve üzeri olanlar zengin olarak değerlendirilebilir. O halde verilerin sınıflandırılması ve değerlendirilmesi işlemi tamamen sezgiseldir. İçinde bulunulan zamana ve değerlendirmeyi yapan kişiye göre değişiklikler gösterebilmektedir. Kesin değerlere dayanan düşünce yöntemi yerine yaklaşık değerleri kullanan bulanık mantık, insan düşüncesine yatkın olması nedeniyle şimdiye kadar birçok sistemde kullanılmıştır. Bulanık mantığın uygulanacağı sistemlerin davranışının kurallar ile ifade edilebiliyor olması ve karmaşık matematiksel işlemler gerektirmiyor olması gerekmektedir. Bunun aksi bir sistemde bulanık mantık etkin çözümü sağlamayacaktır ve büyük olasılıkla beklenen değerleri vermeyecektir [1].

Veri tabanı sistemlerini düşündüğümüzde, verilerin kurallar ile değerlendirilebildiği basit sayısal ve mantıksal işlemler ile sorgulanabildikleri görülmektedir. O halde mevcut bir sistemde, kuralları sizin koymanız, verilerin istediğiniz doğrultuda değerlendirilmesini ve karar alma mekanizmasının tamamen sezgileriniz doğrultusunda olmasını sağlamaktadır. Bu noktada bulanık mantık sistemlerde, dikkat edilmesi gereken bir diğer önemli nokta, sistemin düzgün çalışabilmesi için kuralları belirleyen kişinin uzman kişi olmasıdır. Ancak bu şekilde elde edilen sonuçlar sağlıklı ve gerçeği yansıtmakta başarılı olabilirler. Son yıllarda bulanık sorgular üzerine yapılan çalışmaların sayısında artış gözlenmektedir [2], [3], [4], [5]**.** Bu çalışma kapsamında klasik veri tabanları üzerinde bulanık

sorgulama gerçekleştiren bir sorgulama arayüzü geliştirilmiştir.

2. Neden Bulanık Sorgu?

Bu sorunun yanıtı: "bulanık sorgular ile klasik sorgu cümleleri ile elde ettiğimiz sonuçlardan daha doğru sonuçlar alınabilmektedir ve ihtiyacımız olan verilerin tümüne erişim sağlanabilmektedir." şeklinde olabilir. Bir karar alırken, tüm gerçeklere erişmeyi ve bu gerçeklerin doğruluk derecesinin olabildiğince yüksek olmasını isteriz. Bu anlamda klasik sorgular ile elde edilen bilgiler karar mekanizmasının doğru çalışmasını tam anlamıyla sağlayamaz. Çünkü klasik veri tabanı sorgulamalarında belirlenen kesin sınırlar dışında kalan kayıtların durumları gözlenemez. Sorgu sonuçlarında sadece seçilmiş kritere birebir uygunluk gösteren kayıtlar gözlenebilir. Bu seçimlerin yapılabilmesi için klasik mantıktaki karşılaştırma ifadeleri kullanılır. Uygulamada, genellikle kullanılan sorgulama dili, ilişkisel veri tabanları için standart hale geldiğinden beri SQL dilidir [6]. SQL, sorgular için kesin kurallar ve komutlar içeren bir dildir. Sorgu sonuçları, sorgudaki kriterleri sağlayan kayıtlardan oluşmaktadır. Bir klasik sorgu kriteri, mantıksal ve matematiksel ifadelerden oluşmaktadır.

Klasik sorgulamaların getirdiği kısıtlamalara, esnek ve akıllı sorgular çözüm olarak sunulmuştur. Bulanık mantık, insan düşüncesine olan yatkınlığı ve bulanık ifadeleri işleyebilmesi ile çözüme en uygun yöntem olarak belirlenmiştir [7].

Yukarıda anlatılmak istenenleri bir sorgu üzerinde inceleyelim.

Tablo-1: Laboratuvar Tablosu

Data in Table 'LABORATUAR'

NUMARA	AD	FNOTU	RAPOR	DEVAM	SONUC
1	SEVINC	84	10	9	0,4
2	SERPIL	89	8	7	0,9
3	AYSEL	20	4	2	0
4	ENIS	80	5	3	0
5	ONUR	68	3	8	0
6	HARUN	40	9	9	0
7	OZCAN	83	8	10	0,3
8	GUNAY	30	1	4	0
9	BANU	81	7	9	0,1
10	LEVENT	90	7	2	1
11	NURAY	81	6	7	0,1
12	EYLEM	95	10	9	1

Tablo 1'deki örnek tablo üzerinde sorgulama işlemini gerçekleştirelim. Tablodaki kayıtlardan başarılı öğrencileri elde edelim. 'Notu >=85' ve 'Devam>=8' olan öğrenciler başarılı olarak tanımlanmaktadır.Geleneksel bir sorgu şu şekilde olacaktır :

SELECT NUMARA, AD, FNOTU, DEVAM **FROM** LABORATUAR **WHERE** DEVAM>=8 **AND** FNOTU>=85

Ve sorgu sonucunda seçilen kriterlere kesin uygunluk gösteren kayıtlar listelenir.

NUMARA	AD	FNOTU	DEVAM
2	SERPİL	89	8
12	EYLEM	95	10

Yukarıdaki sonuçlar dikkatlice incelendiğinde, 1 numaralı öğrencinin devam durumu çok iyi olmasına rağmen, final notu 84 olduğu için listede yer almadığı gözlemlenir. Belirlenen kriterler ile kayıt arasında çok küçük fark olmasına rağmen, birbirlerine çok yakın durumda olan kayıtlar farklı sonuç kümelerinde yer alacaklardır. Bu da değerlendirmeyi esneklikten uzaklaştırır ve değerlendirmede kesin kurallar dışına çıkılmasına izin vermez.

Sorgu aralığının genişletilmesi bu probleme çözüm olarak sunulabilir. SQL sorgusunun WHERE bloğunda FNOTU>=83 VE DEVAM>=7 şeklinde bir değişiklik yaptığımız zaman elde edilen kayıtlar aşağıda görülmektedir.

NUMARA	AD	FNOTU	DEVAM
1	SEVINC	84	9
2	SERPİL	89	7
7	OZCAN	83	10
12	EYLEM	95	9

Bu kez de 9 ve 11 numaralı öğrencilerin notları ve devam durumları seçilen kriterlere çok yakın olsalar dahi yine sonuç kümesinin dışında kalmışlardır. 12 kayıtlı bir veri tabanında sorgu aralığını genişletmek, bir çözüm olarak düşünülebilir olsa da; çok daha büyük boyutlu veri tabanlarında bu çözüm yöntemi kabul edilemez. Sorgu aralığını genişletmek karar

alma mekanizmasının daha doğru çalışmasına etki etmeyecektir.

Klasik sorgu işlemcisi ile gerçekleştirmeye çalıştığımız sorguyu, bu çalışmada kapsamında geliştirilen bulanık sorgu işlemcisi ile gerçekleştirmeye çalışalım.

"FNOTU MÜKEMMEL VE DEVAM İYİ" şeklinde başarılı öğrencileri listelemek için oluşturulan bir bulanık sorgu cümlesini değerlendirelim. Bulanık sorgu sonucunda her kaydın hangi sonuç kümesine üye olduğunu sonuç kümelerine olan üyelik dereceleri ile birlikte gözlemleyebiliriz. Böylece başarı durumunu değerlendirebilmek için derece derece farklılık gösteren çok daha geniş bir başarılı öğrenci kümesi elde edebiliriz. Bu şekilde, çok küçük not farklılıklarından dolayı başarılı ya da başarısız diye birbirinden ayrılan öğrencilerin, başarılı kavramına yakınlık durumlarını elde edip, öğrenciler için daha sağlıklı kararlar alabiliriz. Klasik sorguda sadece belirlenen aralıktaki öğrencilerin başarı durumları hakkında bilgi sahibi olabilirken; Şekil-1'de görüldüğü gibi, bulanık sorgu sayesinde veritabanındaki tüm kayıtların başarı durumları hakkında bilgi sahibi olunabilir.

	NUMARA	AD	FNOTU	DEVAM	SONUC
▶	1	SEVINC	84	9	0,4
	2	SERPIL	89	7	0,5
	3	AYSEL	20	2	0
	4	ENIS	80	3	0
	5	ONUR	68	8	0
	6	HARUN	40	9	0
	7	OZCAN	83	10	0,3
	8	GUNAY	30	4	0
	9	BANU	81	9	0,1
	10	LEVENT	90	2	0
	11	NURAY	81	7	0,1
	12	EYLEM	95	9	1

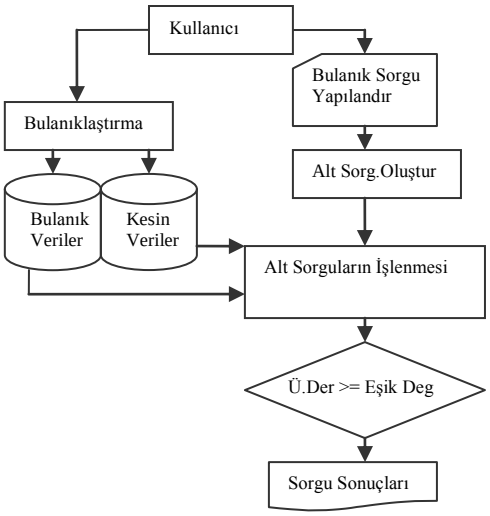
Şekil-1: Bulanık Sorgu Sonuçları

2. Bulanık Sorgu Aracı

İnsan düşünce sistemine yakın olan ve günlük konuşma dilindeki bulanık ifadeler yardımı ile yapılandırılan veri tabanı sorguları "Bulanık Sorgular" diye adlandırılmaktadır. Bulanık sorgularda, klasik sorgulardan farklı olarak sorgu sonuç kümesi, sorguda aranan koşullara ne derece uygun olduklarına göre derecelendirilmiş kayıtlardan oluşur.

Geliştirilen program, Borland Delphi 6.0 ortamında hazırlanmış, veritabanı yönetim sistemi olarak SQL Server 7.0 kullanılmıştır. Yazılım temel olarak, niteliklerin bulanıklaştırıldığı ve bulanık sorgunun gerçekleştirildiği iki bölümden oluşmaktadır.

Bu iki bölüm, 1.2.1 ve 1.2.2' de anlatılmaktadır ve programın genel akış şeması Şekil 2' de gösterilmektedir.



Şekil-2: Programın Akış Şeması

2.1 Niteliklerin Bulanıklaştırılması

Bulanık kümeler, değişik üyelik dereceleri olan elemanların oluşturduğu kümelerdir. Bir bulanık kümenin elemanları, üyelik derecelerinden oluşan bir evrensel küme üzerinde dağılmış durumdadır. Elemanların, aynı evrensel kümede farklı bulanık kümelerine üyelikleri bulunabilir. Yani bulanık olmayan bir değer ancak 1 kümenin elemanı olabilirken, bulanık değer farklı bulanık kümelere üye olabilir.

Bulanık küme elemanları, "üyelik dereceleri" evrensel kümesine teorik fonksiyonlar ile yerleşirler. Örneğin A, bir bulanık kümeyi ifade etsin. Bahsedilen fonksiyonlar, A kümesinin elemanlarının

[0, 1] aralığında değerler almalarını sağlar. Varsayalım x, A bulanık kümesinin bir elemanı ve $\mu_A(x)$, x elemanının A kümesine olan üyelik derecesi olsun. A bulanık kümesi aşağıdaki gibi ifade edilir.

$$\mu_A(x) \in [0,1],$$

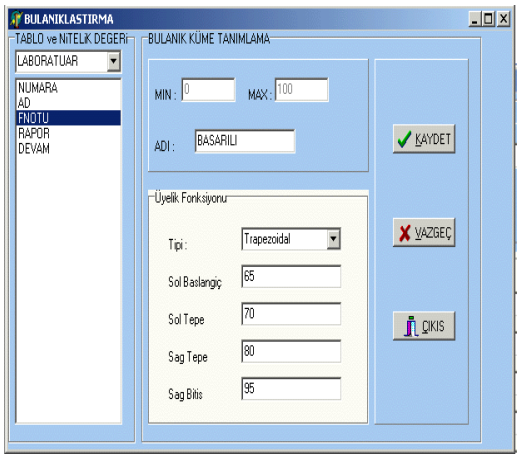
$$A=(x, \mu_A(x) / x \in X)$$

$$A=\sum \mu_A(x_i)/x_i = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \ldots\ldots\ldots\ldots + \mu_A(x_n)/x_n$$

Programda üyelik fonksiyonlarının belirlenmesinde, yukarıdaki denklem kullanılmıştır. Yani üyelik fonksiyonu, eleman ve üyelik derecesi çifti ile belirtilmiştir [8].

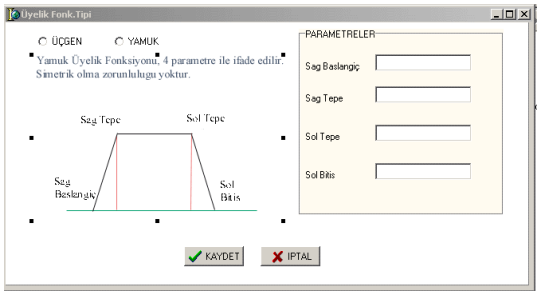
Bulanıklaştırma işlemlerini gerçekleştirmek için, öncelikle veri tabanındaki niteliklere erişmek, üyelik fonksiyonlarını bu nitelikler üzerinde tanımlamak gerekmektedir.

Kullanıcının amacına hizmet edecek bulanık olmayan veriler, "KlasikDB" adındaki veritabanında ilgili tablolarda depolanmış durumdadırlar. Program çalıştırıldığında MS SQL Server'a bağlanarak, daha önceden oluşturulmuş olan "KasikDB" tablolarına erişir. Bu tablolar kullanıcının seçim yapması için listelenir. Üzerinde sorgulama yapılacak tablo seçilir. Yine aynı form üzerinde seçilen tabloya ait nitelik değerleri listelenir. Hangi nitelik değeri bulanıklaştırılacak ise kullanıcı tarafından seçilir. Örneğin, veritabanındaki "LABORATUAR" tablosundaki "FNotu" niteliği üzerinde bulanıklaştırma işlemi gerçekleştirilmesi durumunda, öncelikle bulanık kümenin değer aralığı belirtilmelidir. Örneğin, "FNotu" niteliği için minimum sınır değeri 0; maksimum sınır değeri 100' dür (Şekil 3). Bu aralık saptaması bir kez gerçekleştirilir. Aynı bulanık küme üzerinde başka üyelik fonksiyonları tanımlanırken Min ve Max alanlarının değiştirilmesine program izin vermez. Bir bulanık küme üzerinde kullanıcı istediği kadar aynı isimli üyelik fonksiyonu tanımlayabilir.

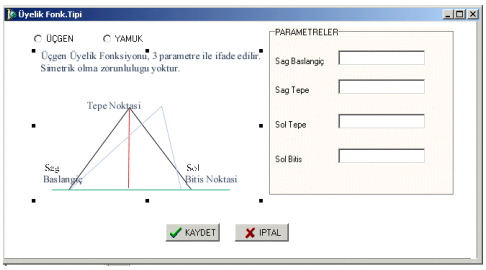


Şekil-3: Bulanıklaştırma Ekranı

Bu işlemden sonra bulanık küme üzerinde tanımlanacak üyelik fonksiyonunun adı ve tipi yani teorik formu belirtilmelidir. Üyelik fonksiyonu Triangular form, Trapezoid form, Exponential form, S-function vb... şeklinde olabilmektedir. Bu uygulamada üyelik fonksiyonu tipi olarak Triangular form (Şekil 4.a) ve Trapezoid form (Şekil 4.b) seçenekleri sunulmuştur. Üyelik fonksiyonları konuşma dilinden alınan ifadeler ile adlandırılır. Örneğin, FNotu bulanık kümesi üzerinde, "Başarısız", "Geçer", "Ortalama", "Başarılı", "Mükemmel" üyelik fonksiyonları tanımlanabilir.



Şekil-4.a: Triangular Üyelik Fonk.Formu



Şekil-4.b: Trapezoidal Üyelik Fonk.Formu

Bulanıklaştırma için gerekli olan alanlar, kullanıcı tarafından eksiksiz doldurulduğu zaman bulanıklaştırma işlemi uygulanan tablo adı, nitelik adı, bulanık kümenin sınır değerleri, üyelik fonksiyonu adı, tipi gibi bulanıklaştırma ile ilgili tüm bilgiler "BulanikBilgiler" veritabanındaki "Bulanikkume" tablosuna eklenir. Bir üyelik fonksiyonu tanımlamak "BulanikKume" tablosuna yeni bir kayıt eklemek demektir.

Bir bulanık küme üzerinde Triangular tipinde bir üyelik fonksiyonu tanımlamak demek üyelik fonksiyonunu oluşturan elemanlara aşağıdaki işlemlerin uygulanması demektir. Benzer işlemler yamuk üyelik fonksiyonu için de uygulanacaktır.

$$\mu_A(x) = \begin{cases} 0 & \ldots\ldots \text{ x<a ya da c<x} \\ \dfrac{x-a}{b-a} & \ldots\ldots a \le x \le b \\ \dfrac{c-x}{c-b} & \ldots\ldots b \le x \le c \end{cases}$$

Sonuç olarak üzerinde sorgulama yapılacak verilerin bulunduğu tablodaki nitelikler üzerinde istenildiği kadar aynı isimleri alan üyelik fonksiyonu tanımlanabilir. Bulanık küme ve üyelik fonksiyonlarının belirlenmesinde kullanıcıya maksimum esneklik sunulmuştur.

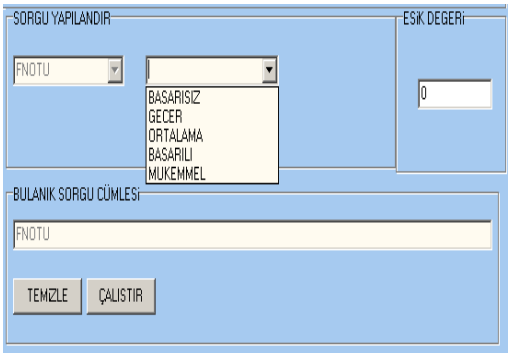
2.2 Bulanık Sorgulama İşlemi

Programın ikinci kısmında bulanık sorgu cümlesi oluşturulur. Sorguya istenen sayıda özellik eklenebilir. Sorgunun doğru formda olup olmadığının anlaşılması için yazımı test edilir.

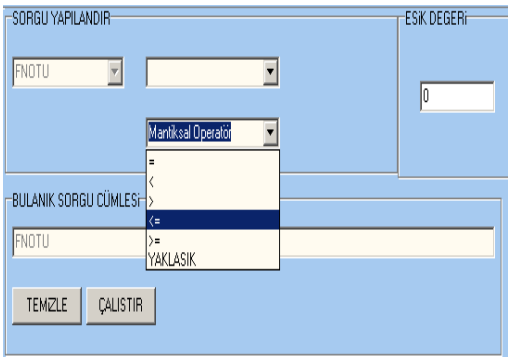
Bulanık sorgunun söz dizimi aşağıdaki gibi olmalıdır.

NİTELİK BULANIKDEĞİŞTİRİCİ ÜYELİKFONKSİYONU DEĞİL **VE/(VEYA) NİTELİK** ÜYELİKFONKSİYONU

Bulanık sorgu cümlesi oluşturulurken daha önceden Bulanıklaştırma kısmında kullanıcı tarafından tanımlanmış olan ve Şekil-5.a'da görülen bulanık kümelerden ya da Şekil-5.b'de görülen mantıksal operatörlerden yararlanılır.



Şekil-5.a: Bulanık Kriterlerin Seçimi



Şekil-5.b: Mantıksal Operatörlerin Seçimi

Oluşturulan bulanık sorgu cümleleri üzerinde bağlaçlar aranır ve bulunanlar dinamik bir dizide tutulurlar. Bulanık sorgu cümlesi bu bağlaçlar ile parçalanarak alt sorgu cümleleri oluşturulur. Oluşan her alt sorgu cümlesi yine dinamik bir dizi yapısında saklanır. Dizi elemanları tek tek programın sintax prosedürüne gönderilir. Üstte belirtilen sorgu formuna uyan alt sorgular "doğru" sonucu ile alt sintax prosedüründen geri döner. Oluşan alt sorgulardan bir tanesi dahi kurala uymaz ise yazılan bulanık sorgu cümlesinin "Hatalı" yazıldığı mesajı kullanıcıya gider ve bulanık sorgu yeniden yazılır.

Tüm alt sorguların yazımı istenen formatta ise yazımı doğrudur. Yazımı doğru formda olan sorgu cümlesi işlenebilir demektir.

Sorgu işleme aşamasında, programın "sorguislem" prosedürüne gelen alt sorgu cümlesinin içerisinde geçen üyelik fonksiyonunun adı araştırılır. Sorgu içinde üyelik fonksiyonunun adını bulmak demek aynı zamanda sorgunun bulanık sorgu olduğunun belirlenmesi demektir. Bu durumda "BulanikKume" tablosundan, bu üyelik fonksiyonuna ait bilgiler alınır. Oluşturulan bulanık küme üzerinde işlem yapabilmek için ihtiyaç duyulan tüm veriler elimizdedir.

"KlasikDB" veri tabanında bulanık olmayan değerleri içeren tablo kayıtları tek tek incelenir. Her kaydın ilgili nitelik değerinin üyelik fonksiyonuna ait üyelik derecesi hesaplanır. Bulunan değer [0,1] aralığındadır ve veritabanındaki kaydın bulanık kümeye üyeliğini verir. Bu işlem her alt sorgu cümlesi için tekrarlanır

Yukarıda anlatılan işlemleri bir örnek üzerinde izleyelim. "LABORATUAR" tablosu üzerinde "FNOTU MÜKEMMEL VE DEVAM İYİ" kuralına uyan öğrencileri bulmak için bir bulanık sorgu cümlesi oluşturulduğunu varsayalım.

Sorgu cümlesi iki alt sorgu cümlesine parçalanır. Öncelikle "FNOTU MÜKEMMEL" alt sorgusundaki nitelik değeri ve üyelik fonksiyonu aranır. Bulanık olmayan değerlerin saklandığı "LABORATUAR" tablosundaki "FNOTU" alanında saklanan değerlerden bir kayıt kümesi oluşturulur. Bu kayıt kümesindeki her değerinin "MÜKEMMEL" bulanık kümesine olan üyelik derecesi hesaplanır. Bunu gerçekleştirebilmek için bulanık bilgi tabanının yer aldığı "BulanıkBilgiler" tablosundan Klasik tablo adi "LABORATUAR" ve üyelik fonksiyonu adi "MÜKEMMEL" olan kayda ait bilgiler çekilir. Üyelik fonksiyonunun tipi için tanımlanmış fonksiyon göz önünde bulundurularak kaydın, üyelik fonksiyonuna ait üyelik derecesi bulunur. [0..1] aralığındaki bu değer, bulanık olmayan değerlerin tutulduğu tablodaki ilk kaydın "FNOTU MÜKEMMEL" alt sorgusuna uygunluk derecesidir. Aynı yöntem ile tüm kayıtların sıra ile "FNOTU MÜKEMMEL" alt sorgusuna olan uygunluk dereceleri hesaplanır. Bu işlemlerin aynısı "DEVAM İYİ" alt sorgusu için de tekrarlanır.

Kayıtların sorgulara uygunluk dereceleri iki boyutlu dinamik bir dizi yapısında saklanır. Dizinin ilk boyutu tablodaki kayıtların indisini; ikinci boyutu hangi alt sorgu cümlesine uygunluk derecesinin araştırıldığını yani alt sorgu cümlelerinin indisini tutar. İki boyutlu dinamik bir dizide tüm üyelik dereceleri saklanmış olmaktadır. (Tablo 2)

Tablo-1: Dinamik Dizi(n kayıtlı bir veritabanı, m parçalı bir bulanık sorgu için)

	1.Alt Sorgu	**2.Alt Sorgu**	**.....**	**m.Alt Sorgu**
1. Kayıt	$\mu_{1,1}$	$\mu_{1,2}$		$\mu_{1,m}$
2.Kayıt	$\mu_{2,1}$	$\mu_{2,2}$		$\mu_{2,m}$
.....				
n.Kayıt	$\mu_{n,1}$	$\mu_{n,2}$		$\mu_{n,m}$

Tablo-1: Bağlaç Dinamik Dizisi

VE	VEYA	.....	VE

Böylece tablodaki her kayıt için 0'dan n'e kadar olan alt sorgulara uygunluk derecesi bilinmektedir.

Aynı şekilde sorgu cümlesinde yer alan bağlaçlar da dinamik biri diziye yerleştirilirler.

Bağlaç dizisinin "i". elamanı "VE" ise Üyelik derecelerinin saklandığı dizinin $\mu_{j,i-1}$ ve $\mu_{j,i+1}$ elemanlarının sayı değerleri, yani "j". elemanın (i-1) ve (i+1). alt sorgulara uygunluk derecesi "Min" işlemine tabi tutulur. Bağlaç dizisinin "i". elamanı "VEYA" ise $\mu_{j,i-1}$ ve $\mu_{j,i+1}$ elemanlarının sayı değerleri "Max" işlemine tabi tutulur.

$$\text{Sonuc}=\min\left(\mu_{Sorgu1}(x), \mu_{Sorgu2}(x)\right)$$

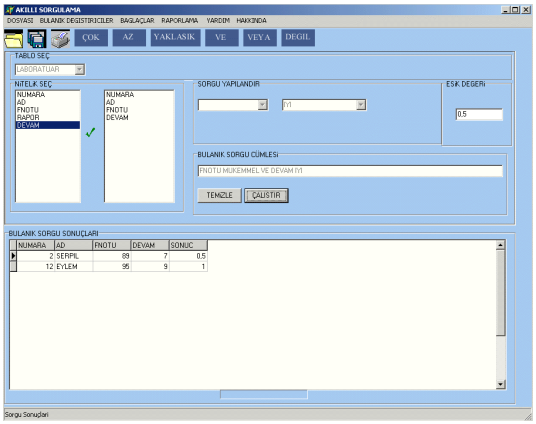
(1. ve 2. alt sorgular "VE" bağlacı ile bağlanmış ise),

$$\text{Sonuc}=\max\left(\mu_{Sorgu1}(x), \mu_{Sorgu2}(x)\right)$$

(1. ve 2. alt sorgular "VEYA" bağlacı ile bağlanmış ise),

Kayıtların sorguya toplamda uygunluk derecesini bulurken, öncelik “VE” işlemindedir. Daha sonra bulunan sonuçlar ile yenilenen dizide “VEYA” işlemi uygulanır.

Bu şekilde veri tabanındaki her kayıt için bulanık sorguya uygunluk derecesi hesaplanmış olur. (Şekil 6)



Şekil-6: Bulanık Sorgu Sonucu Ekranı

Programda oluşturulan sorgu cümlesinin bulanık sorgu olmadığı tespit edildiğinde klasik yöntemler ile istenen koşullara uyum sağlayan kayıtların üyelik derecesi 1, uymayan kayıtların üyelik derecesi ise 0 olarak hesaplanır. Program bulanık sorgu ve klasik sorguyu aynı cümlede gerçekleştirebilmektedir.

2.2.1 Dilsel Değiştiriciler

Değiştirici fonksiyonları,[0,1]’den [0,1]’e tanımlı olup, bulanık küme üyelik fonksiyonlarına uygulanabilir. Bunun için daha önceden belirlenmiş modeller “çok”, ”az”, ”yaklaşık”, “az çok” vb. kullanılır [9][10].

Bu uygulama kapsamındaki sorgulama modelinde, sorgulama konuşma diline yakın bir kullanıma olanak verecek biçimde geliştirilmiştir. Bunun için “çok”, ”az”, ”yaklaşık” dilsel değiştiricileri kullanılmıştır.

Eğer bulanık sorgu “ NİTELİK DEĞİŞTİRİCİ ÜYELİK FONKSİYONU ”, kısaca “A is m MF” şeklinde ise; bu ifadede a, veri tabanındaki ilgili niteliğin değerini; m sorguda kullanılan dilsel değiştiriciyi, MF de bulanık kümeyi ifade etmektedir.

O zaman Değişikliğe uğrayan kural aşağıdaki şekildedir.

$$K = \mu_m(\mu_{MF}(x))$$

$\mu_m, \mu_{MF}(x)$ üyelik derecesinin, m değiştirici fonksiyonuna olan üyelik derecesidir.

Örneğin “A1 is m1 MF1 AND / OR A2 is m2 MF2” şeklindeki bir sorgu cümlesi için yapılacak işlem şu şekildedir.

K=min $(\mu_{m1}(\mu_{F1}(x), \mu_{m2}(\mu_{F2}(y))))$, bağlaç VE ise

K=max $(\mu_{m1}(\mu_{F1}(x), \mu_{m2}(\mu_{F2}(y))))$, bağlaç VEYA ise

Örneğin, bulanık sorgu cümlesi “NOTU ÇOK YÜKSEK DEĞİL ” şeklinde olsun.

1. kayıt için, ilk sorgunun uygunluk derecesi : ud olduğu durumda;

ÇOK değiştiricisinin sorguya etkisi: ÇOK(ud)= ud^2, şeklinde hesaplanır.

DEĞİL değiştiricisinin sorguya etkisi:

DEĞİL (ud^2) =1 - ud^2 , şeklinde hesaplanır.

Bulanık Sorgu cümlesi “NOTU YAKLAŞIK 80”

1. kayıt için, ilk sorgunun uygunluk derecesi “ud” olduğu durumda

YAKLAŞIK değiştiricisinin bulanık sorguya etkisi şu şekilde hesaplanır.

Veri tabanındaki bulanık olmayan değerler arasında NOTU değişkenin aldığı minimum ve maximum değerler ve veri tabanındaki kayıt sayısı bulunur.

$\varepsilon =$ (maxNOT-minNOT)/KayıtSayısı

$80-\varepsilon \leq NOTU \leq 80+\varepsilon$

aralığında olmayan kayıtların sorguya uygunluk dereceleri 0’dır.

Bu aralıktaki bir NOTU değeri için ise uygunluk derecesi : $1-\left(\left|80-NOTU\right|/\varepsilon\right)$ kadardır.

Yukarıda bahsedilen fonksiyonlar, bulanık sorgu işlemede en son aşama olarak, bulunan üyelikderecelerine uygulanırlar.

2.2.2 Eşik Değeri (Bölüm Kesmesi)

Bulanık A kümesinin $\alpha - Cut$ kümesi, A_α ile gösterilir ve X evrensel kümesinin A kümesindeki bütün elemanlarında üyelik derecesi α özel değerinden büyük ve eşit olanları içerir [1].

$$A_\alpha = \{x \in X | \mu_A(x) \geq \alpha\}$$

Programda, bulanık sorgu cümlesi oluşturulduktan sonra, sorgu sonuçları için α bölüm kümesi oluşturulmak isteniyorsa; α kesme değeri girilmelidir.

Tüm bu işlemlerden kullanıcının seçimi doğrultusunda oluşturulan bulanık sorgulamalar, değerlendirilmiş ve kayıtların sorguya üyelik dereceleri istenen eşik değeri belirtilerek listelenmiştir.

Örneğin, (Şekil 6) 'da " FNOTU MÜKEMMEL VE DEVAM İYİ " sorgusunun sonuçları eşik değeri 0,5 seçilerek listelenmiştir.Bu durumda sadece 0,5 ve üzeri derecesinde sorguya uyan kayıtlar listelenecektir.

3. Sonuç

Bu çalışmada, klasik veri tabanlarında bulanık sorgulama gerçekleştirebilen bir yazılım aracı tanıtılmıştır. Temelde iki bölümden oluşan programda bulanıklaştırma kısmında veri tabanı üzerindeki kullanıcının belirlediği herhangi bir alan üzerinde bulanık kümeler oluşturulmaktadır. Bulanık sorgunun gerçekleştiği ikinci kısımda ise, günlük konuşma dilindeki ifadeler ile bulanıklaştırılan nitelikler yardımı ile veri tabanı kayıtları sorgulanmaktadır. Bu çalışmada, klasik sorgulamanın getirdiği kısıtlamalardan sıyrılıp, daha esnek sorgulama yapılmasına olanak sağlayacak bir sorgulama dili geliştirmek hedeflenmiştir. Sonuç olarak sezgisel sorgulama yapılmasına olanak sağlayan, sorgu sonuçlarını derecelendirerek gösteren ve bu şekilde daha sağlıklı kararlar alınmasına katkı sağlayan bir bulanık sorgulama aracı geliştirilmiştir.

Kaynakça

[1] **Elmas, Ç.,** 2003 Bulanık Mantık Denetleyiciler, Seçkin, Sıhhiye, Ankara, 63

[2] **Rasmussen, D.,Yager, R.R.**, 1997. Fuzzy Tool For Datamining, Intelligent Data Analysis,Vol.1,Elsevier Science Inc.

[3] **Bosc, P.,Pivert, O.**, SQLf: 1997. A Relational Database Language for Fuzzy Querying, IEEE Transactions on Fuzzy Systems, Vol. **3**.

[4] **Cox, E.**, 2000. FuzzySQL-A Tool for Finding The Truth-The Power of Approximate Database Queries, PC AI-Intelligent Applications,Vol 14.

[5] **Eminov, M.**, Querying Database by Fuzzification of Attribute Values, http://idari.cu.edu.tr/sempozyum/bil46.htm

[6] **Elmasri, R.**, **Navathe, S., B.**, 2000. Fundamentals of Database Systems, Addison Wesley.

[7] **Zadeh, L.A.**, 1965. Fuzzy Sets. Information and Control, 8, 338-353.

[8] **Jamshidi, M.**, 1993. Fuzzy Logic and Control: Software and hardware Applications, PTR Prentice-Hall, Englewood Cliffs, New Jersey., 16-18.

[9] **Wang Li-Xin.**, 1994. Adaptive Fuzzy Systems and Control: Desingn and Stability Analysis, PTR Prentice-Hall, Englewood Cliffs, New Jersey, 9-14.

[10] **Bosc, P.,Prade H.**, 1997. An Introduction To the Fuzzy Set Possibility Theory-Based Treatment of Soft Queries and Uncertain or Imprecise Databases, Uncertainty Management in Information Systems: From Needs to Solutions, Kluwer Academic Publ., 285-324