

**İLİŞKİSEL VERİ TABANLARINDA
FONKSİYONEL VE DERECELİ
BULANIK İLİŞKİ SORGULAMA ARAÇI**

YÜKSEK LİSANS TEZİ

Mat. Müh. Erdiñ Fadıl ERTANİŞ

(509971409)

Anabilim Dalı: Mühendislik Bilimleri

Program: Sistem Analizi

Tez Danışmanı: Prof. Dr. Gazanfer ÜNAL

OCAK 2003

ÖNSÖZ

Ömrüm boyunca bana verdikleri sonsuz sevgi ve emek için anne, babama, kadeşlerime Dilekile Ömür'e minnetlerini sunuyorum..

Filiz senteşvik etmeseydi ne belki hiç başlamayacaktım Hep yanımda oldun. Sana çok şey borçluyum Sağol...

Sevgili dostlarımla Barış KIZILSAÇ ve Mehmet Emin ŞALVA Dostluğunuz ,yardımlarınız ve anlayışınız için teşekkür ederim

Yüksek lisans eğitiminin her aşamasında manevi desteğini hissettiren , kişisel gelişimimde önemli payı olan Sayın Haluk BERTAN'a teşekkürlerini sunuyorum

Lisans ve yüksek lisans öğrenimim boyunca desteğini esirgemeyen, ufuk açan Sayın hocam Prof. Dr. Gazanfer ÜNAL'a şükranlarını sunarım

Ocak 2003

Erdiñ Fadıl ERTAN Ş

İÇİNDEKİLER

ÖNSÖZ	ii
TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
ÖZET	viii
SUMMARY	x
1. GİRİŞ	1
1.1 Veri Madenciliği	2
1.1.1 Veri Anbarları ve Veri Madenciliği	2
1.1.2 Veri Madenciliğinin Örnek Uygulamaları	3
1.1.3 Bilgi Keşfi Süreci	5
1.1.4 Veri Madenciliğini Etkileyen Faktörler ve Uygulamanın Getirileri	7
1.2 Uzman Sistemler	7
1.2.1 Uzman Sistemin Bileşenleri	8
1.2.2 Uzman Sistem Kullanımının Faydaları	9
2. BULANIK KÜMELER VE BULANIK MANTIK	11
2.1 Bulanık Kümeler	11
2.1.1 Karakteristik Fonksiyonun Özellikleri	12
2.1.2 Bulanık Kümelerin Özellikleri	15
2.1.3 Temel Bulanık Küme İşlemleri	16
2.1.4 Bulanık Kümelerin Özellikleri	18
2.2 Bulanık Mantık	18
2.2.1 Doğruluk Değeri	19
2.2.2 Temel Bulanık Mantık İşlemleri	19
2.2.3 Dilsel Pekiştiriciler	20
3. VERİ TABANLARINDAKİ BULANIK BAĞLIILIKLAR	22
3.1 Bulanık Fonksiyonel Bağlılık ve Dereceli Bulanık Bağlılık	22
3.2 Dilsel Özetleme	23
3.3 Bulanık Fonksiyonel Bağlılıklar (FFD)	25

3.4	Dereceli Bulanık Bağımlılıklar (GFD)	28
4	GELİŞTİRİLEN UYGULAMA	32
4.1	Kullanılan Algoritmalar ve Akış Diyagramları	32
4.1.1	Bulanık Fonksiyonel İlişki Hesabı için Algoritma	32
4.1.2	Dereceli Bulanık İlişki Hesabı için Algoritma	34
4.2	Yazılımda Uygulanan Bulanık Fonksiyonlar	35
4.3	Geliştirme Ortamı ve Kullanılan Temel Yazılım Yapıları	38
4.3.1	Kullanılan Temel Veritabanı Birimleri	39
4.3.2	Kullanılan Temel PL/SQL Yapıları	39
4.4	Geliştirilen Birimler	41
4.4.1	Uygulamanın Oluşturduğu Tablolar	42
4.5	Dereceli ve Fonksiyonel Bulanık İlişki Sorgulama Aracının Kullanımı	42
4.5.1	Genel	43
4.5.2	Fonksiyonel Bulanık Sorgulama	44
4.5.3	Dereceli Bulanık İlişki Sorgulama	47
4.5.4	Geçmiş Sorguların Takibi	50
5	2002 YILI TÜRKİYE DEPREMLERİ İLE GERÇEKLEŞTİRİLEN VERİ MADENCİLİĞİ ÇALIŞMASI	54
5.1	Analiz Edilen Verinin Veri Tabanında Saklanması	54
5.2	Deprem Verisinin Gruplanması	54
5.3	Doğruluk Değeri Sorgulanan Önermeler	55
5.4	Veri Madenciliği Çalışmasının Değerlendirmesi	62
5.4.1	Doğruluk Değeri Yüksek Olan Önermelere İlişkin Yorumlar	62
6	SONUÇLAR VE ÖNERİLER	65
	KAYNAKLAR	67
	EK A	68

ÖZGEÇMİŞ..... 73

TABLO LİSTESİ

Tablo 2.1:	X de tanımlı bulanık kümeler ve üyelik değerleri	14
Tablo 3.1:	FFD hesaplaması için örnek veritabanı	26
Tablo 3.2:	FFD hesaplaması için gerekli ara hesaplamalar	26
Tablo 3.3:	GF D hesaplaması için gerekli ara hesaplamalar	30
Tablo 5.1:	Alanlar için kullanılan eşik değerleri	55
Tablo 5.2:	39. Paralelin kuzeyinde meydana gelen magnitud > 4.0 dan depremler	56
Tablo 5.3:	39. Paralelin kuzeyinde meydana gelen magnitud $\in [3.0, 4.0)$ dan depremler	56
Tablo 5.4:	Paralelin kuzeyinde ve 32. Enlemin doğusunda meydana gelen magnitud > 4.0 dan depremler	57
Tablo 5.5:	Paralelin kuzeyinde ve 32. Enlemin doğusunda meydana gelen magnitud $\in [3.0, 4.0)$ dan depremler	57
Tablo 5.6:	39. Paralelin kuzeyinde ve 32. Enlemin batısında meydana gelen magnitud > 4.0 dan depremler	58
Tablo 5.7:	39. Paralelin kuzeyinde ve 32. Enlemin doğusunda meydana gelen magnitud $\in [3.0, 4.0)$ dan depremler	58
Tablo 5.8:	39. Paralelin güneyinde meydana gelen magnitud > 4.0 dan depremler	59
Tablo 5.9:	39. Paralelin güneyinde meydana gelen magnitud $\in [3.0, 4.0)$ dan depremler	59
Tablo 5.10:	39. Paralelin güneyinde ve 32. Enlemin doğusunda meydana gelen magnitud > 4.0 dan depremler	60
Tablo 5.11:	39. Paralelin güneyinde ve 32. Enlemin doğusunda meydana gelen magnitud $\in [3.0, 4.0)$ dan depremler	60
Tablo 5.12:	39. Paralelin güneyinde ve 32. Enlemin batısında meydana gelen magnitud > 4.0 dan depremler	61
Tablo 5.13:	39. Paralelin güneyinde ve 32. Enlemin doğusunda meydana gelen magnitud $\in [3.0, 4.0)$ dan depremler	61

ŞEKİL LİSTESİ

Şekil 1.1:	Veri Tabanları ve Veri Ambarları	2
Şekil 1.2:	Analiz için Kullanılacak Verinin Hazırlanması	5
Şekil 1.3:	Uzman Sistemin Öğeleri	8
Şekil 2.1:	F Bulanık Kümesinin Dagramüzerinde gösterimi	12
Şekil 2.2:	Geleneksel küme için karakteristik fonksiyon (uzun boylular kümesi)	13
Şekil 2.3:	Bulanık küme için karakteristik fonksiyon (uzun boylular kümesi)	13
Şekil 2.4:	Bulanık kümenin bazı temel özellikleri	15
Şekil 2.5:	Bulanık kümenin değli	16
Şekil 2.6:	Bulanık birleşme için üyelik fonksiyonu	17
Şekil 2.7:	Bulanık kesişme için üyelik fonksiyonu	18
Şekil 4.1:	Bulanık Fonksiyonel İlişkinin Doğruluk Değeri Hesabı Akış Dgr m	33
Şekil 4.2:	Dereceli Bulanık İlişkinin Doğruluk Değeri Hesabı Akış Dagram	34
Şekil 4.3:	Bulanık Fonksiyonel İlişki Sorgu Sonuçlarının Kaydedildiği Tablo	42
Şekil 4.4:	Dereceli Bulanık İlişki Sorgu Sonuçlarının Kaydedildiği Tablo	42
Şekil 4.5:	Uygulama Giriş Ekranı	43
Şekil 4.6:	Fonksiyonel bulanık ilişki sorgulama ekranı	44
Şekil 4.7:	”Tablo” alanı için değer listesi	45
Şekil 4.8:	”Alan1” alanı için değer listesi	45
Şekil 4.9:	Benzerlik fonksiyonunun seçilmesi	46
Şekil 4.10:	Doğruluk değerinin görüntülenmesi	47
Şekil 4.11:	Dereceli bulanık ilişki sorgulama ekranı	48
Şekil 4.12:	Şart alanı için durum operatörü seçilmesi	48
Şekil 4.13:	Where ifadesinin test edilmesi	49
Şekil 4.14:	Sorgu sonucunun görüntülenmesi	50
Şekil 4.15:	Geçmiş bilgilerini temizlenmesi	51
Şekil 4.16:	Geçmiş bilgilerinin dosyaya çıkarılması	51
Şekil 4.17:	Fonksiyonel bulanık ilişki sorgusu için geçmiş dosyası	52
Şekil 4.18:	Dereceli bulanık ilişki sorgusu için geçmiş dosyası	53
Şekil 5.1:	Deprem Verilerini Kaydedildiği Tablo	54

İLİŞKİSEL VERİ TABANLARINDA FONKSİYONEL VE DERECELİ BULANIK İLİŞKİ SORGULAMA ARACI

ÖZET

Yaşadığımız dünyada, süregelen süreçlerin he men he men tümü ilişkisel veritabanı sistemlerinde takip edilir ve arşivlenir hale gelmiştir. Veritabanı sistemleri genellikle, önceden planlanan süreçlerin takip edilmesi ve raporlanması amacıyla kullanılır. Veritabanlarındaki çok değerli nitelikteki pek çok bilgi uygun şekilde sorgulanmadığından kullanılmamaktadır. İlgili sistemler üzerinde yapılacak verimlilik çalışmaları ile, çok değerli çekirdek bilgiye ulaşmak mümkün olabilmektedir.

Bu yüksek lisans tez çalışmasında, ilişkisel veritabanları üzerinde fonksiyonel ve dereceli bulanık ilişkileri sorgulayan ve sonuçlarını saklayan bir uygulamanın geliştirilmesi hedeflenmiştir. Fonksiyonel ilişkilere örnek olarak

“x gıdasını tüketenler, y hastalığına yakalanırlar”

ve dereceli ilişkilere örnek olarak

“boyu uzun olanların, ağırlıkları da fazladır.”

gösterilebilir. Geliştirilen yazılımın doğruluk değeri hesaplama algoritması Dan Rasmussen ve Ronald R. Yager tarafından önerilen doğruluk değeri hesaplama yöntemine dayandırılmıştır. Geliştirilen uygulama ile Oracle tabanlı herhangi bir veritabanındaki tablolar üzerinde fonksiyonel ve dereceli ilişki sorgulaması yapılabilir. Geliştirme çalışması, Oracle 8i Personal Edition veritabanı ve Oracle Developer 6 uygulama geliştirme aracı ile yapılmıştır.

Birinci bölümde, geliştirilen uygulamanın kullanılabileceği disiplinler olan verimlilik ve uzman sistem geliştirme hakkında temel bilgi verilmiştir.

İkinci bölümde, uygulamada kullanılan bulanık küme ve bulanık mantık kavramları hakkında açıklayıcı bilgiler sunulmuştur.

Üçüncü bölümde, kullanılan doğruluk değeri hesaplama sürecinde kullanılan algoritmik yapı açıklanmıştır.

Dördüncü bölümde ise, geliştirme süreci hakkında bilgi verilmiş ve geliştirilen uygulama tanıtılmıştır.

Beşinci bölümde 2002 yılı içerisinde meydana gelen deprem verileri ile yapılan verimlilik çalışması sunulmuştur.

Altıncı bölümde sonuçlar ve önerilere değinilmiştir.

TOOL FOR CALCULATING TRUTH VALUES OF FUZZY FUNCTIONAL AND GRADUAL FUNCTIONAL DEPENDENCIES

SUMMARY

Almost all the procedures of today's life are controlled and stored on relational database systems. In most cases, the database systems only control the actual processes and generate predesigned reports. The valuable core information can be extracted from these database systems by means of data mining studies.

Main purpose of this thesis is to develop a tool for calculating truth values of fuzzy functional dependencies (FFD) and gradual functional dependencies (GFD). An example form of a FFD can be,

“people who uses product x, has disease y.”

and an example form of a GFD can be,

“the taller the persons are, the higher weight they have.”

The truth value calculating algorithm used in the software is the one which was published by Dan Rasmussen and Ronald Yager in 1996. The development Oracle 8i Personal Edition database ve Oracle Developer 6 development tool is used during the development phase.

Contents of the parts are as follows

Part 1 : General explanation about data mining and expert systems

Part 2 : Fuzzy sets and fuzzy logic

Part 3 : The algorithm of calculating GFD and FFD

Part 4 : The tool development

Part 5 A sample data mining study on earthquakes which occurred in Turkey in 2002

Part 6 : Results and comments

1. GİRİŞ

Elektronik alanında 20 nci yüzyılın son üç on yılında meydana gelen teknolojik devrim ve buna bağlı olarak bilgisayar sistemlerindeki inanılmaz hızdaki gelişim günümüz dünyasındaki tüm süreçlerin bilgisayar merkezli bir yapıya dönüşmesine imkan sağlamaktadır. Ucuzlayan, işlem hızları ve veri saklama kapasiteleri artan sistemler daha işlevsel hale gelmiş, kendilerine bir çok ortamda, daha fazla uygulama alanı bulabilmişlerdir. Önceleri askeri ve bilimsel amaçlarla kullanılan bilgisayarlar zamanla iş yerlerinde, okullarda ve evlerde hak ettikleri yeri edinmişlerdir.

Verilerin bilişim sistemlerinde saklanması ve işlenmesinin sonucu olarak yeryüzündeki ulaşılabilir veri miktarı her yirmi ayda kendini iki katına çıkarabilir hale gelmiş, ve veriler etkin şekilde sorgulanabilir hale gelmiştir. Bunun yanı sıra, bilgisayar ağlarındaki veri taşıma kapasitesinin artması ve güvenlik algoritmalarının etkinleşmesi saklanan verilerin uzaktaki sistemler tarafından da kullanıma imkan sağlamaktadır.

Gelişen bilgi işlem teknolojileri ile hayatımızda kullanılan sistemler bir çok güçlü özelliğe sahip olmuşlardır. Örneğin, önceleri sadece hesaplama ve resim formatlama işlevine sahip kasatipci hazırları son yıllarda kullanılan örneklerde olduğu gibi her bir satış kalemini depolayabilir hale gelmiştir. Yapılan her satış hareketinin hangi tarihte yapıldığı, ve bu hareketin hangi müşteriye yapıldığı (kredi kartı ile işlem yapılması durumunda) saklanabilmektedir.

Bilgisayarların veri depolama işlevinin yanı sıra, geliştirilen yazılımlar vasıtasıyla insansı özellikler de edinmektedirler. Yürütülebilmesi için bir çok uzmanın bilgi ve gözetimine ihtiyaç duyulacak olan bir çok sistem bilgisayar teknolojileri yardımıyla en az uzmanın istihdamı ile çalıştırılabilir hale gelmiştir.

Bu yüksek lisans tez çalışmasında günümüz bilişim sistemleri üzerinde kullanılabilecek bir bulanlık ilişkisi sorgulama aracı yazılması hedeflenmiştir. Geliştirme sırasında, Dan Rasmussen ve Ronald R Yager tarafından önerilen ilişki tespit algoritması kullanılmıştır.

Bölüm 1 'de bulanlık sorgulamanın potansiyel uygulama alanı olan verim denciliği ve uzman sistemler hakkında temel bilgiler sunulmuştur. Bölüm 2' de yazılım

algoritmasının temel fikrini oluşturan bulanık küme ve bulanık mantık teorisi ile ilgili bilgiler verilmiştir. Bölüm 4’te geliştirilen uygulama tanıtılmış, ve Bölüm 5’te çalışmanın sonuçları hakkındaki düşünceler ifade edilmiştir.

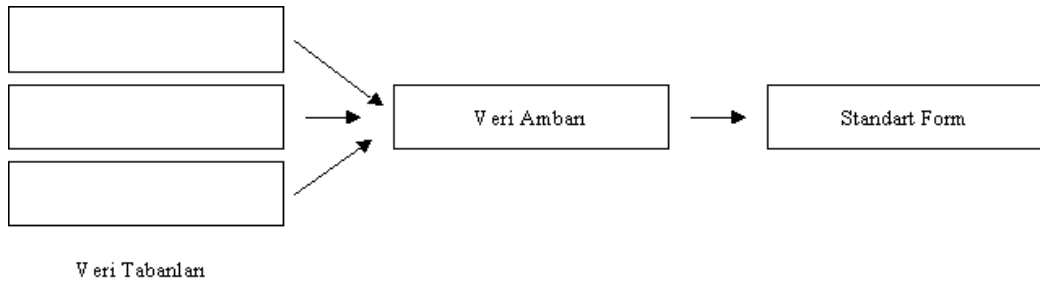
1.1 Veri Madenciliği

Etkin veri depolama ve işleme her zaman etkin bilgi kullanım anlamına gelmemektedir. Çoğu zaman saklanan verinin çeşitliliği, miktarı ve düzensizliği veritabanı sistemlerinden kullanılabilir değerli bilginin elde edilmesine engel olmaktadır. Kısıtlı hacimdeki veriden anlamlı sonuçlar insan emeği çıkarılabilir, ancak bahsi geçen hacimdeki verinin anlamlandırılması için bu amaca yönelik tasarlanan sistemlere ihtiyaç duyulmaktadır.

Veri Madenciliği, bilgisayar programları aracılığıyla yüksek hacimli verinin düzenlenmesi, anlamlandırılması ve gelecekle ilgili tahmin yapmamızı sağlayacak değerli bilgiyi elde etme prosesisidir. Yakın geleceğin geçmişten çok farklı olmayacağı kabul edilirse, eldeki verinin anlamlandırılması ve gelecek stratejilerin belirlenmesinde destek birimi olarak kullanılması anlamlı olacaktır.

1.1.1 Veri Anbarları ve Veri Madenciliği

Veri tabanı sistemleri, iş süreçlerinin takip edildiği sistemlerde en önemli unsur olarak gözümüze çarpmaktadır. OLTP sistemler olarak bilinen bu sistemler genellikle karışık ilişkiler içermektedirler. Verinin saklanması biçimi ilişkisel veri sistemlerinin Karar Destek sistemi olarak kullanılmasını güçleştirir. Hedeflenen verinin sistemden alımı bir çok sorgu, hesaplama ve tekrar gerektirmektedir.



Şekil 1.1: Veri Tabanları ve Veri Anbarları

Veri analarları karar destek süreçlerinde kullanılacak veri yi ilgili işletmenin bir çok veri tabanı sisteminden toplayarak, kullanıcının rahatlıkla anlayacağı biçimde depolayan ve raporlayan sistemlerdir.

Veri analizinde oluşturulan bilgi gözle analiz edilebilecek biçime gelmiş veridir. Analiz işlemlerinde genellikle OLAP(Online Analytical Processing) yazılımları kullanılır. Bu programlar, analistin veriye her alanı bir boyuta karşılık gelen çok boyutlu sanal küpler olarak bakmasını ve analiz etmesini sağlar. Bu sayede boyut bazında gruplama, boyutlar arasındaki korelasyonları incelemeyi kolaylaştırır.

Veri Madenciliğinde hedeflenen, analiz yapan kullanıcının elde edilen sonuçlara katkısını en az düzeye indirmek ve her türlü ilişkinin tespit edilebilmesini sağlamaktır. OLAP programlarında elde edilecek sonuçlar genellikle kullanıcının sistemden sorguladığı durumlarla sınırlıdır. OLAP sistemler bir marketin fatura kayıtlarındaki kullanıcının aklına gelebilecek ilişkileri farkedemezken, veri madenciliği çalışmaları ile bu ilişkiler ortaya çıkarılabilir. Genel olarak veri madenciliği çalışmaları bu tip bağlantıların keşfinde önemli faydalar sağlar.

1.1.2 Veri Madenciliğinin Örnek Uygulamaları

Veri Madenciliği ve veri analarlığı tıp eğitimi iş, elektronik ticaret alanlarında temel ve önemli bir araştırma disiplini haline gelmektedir. Genel anlamıyla eldeki yapılandırılmış veriden belirli amaçya yönelik anlamı ve kullanışlı bilgiyi çıkarmak için uygulamaya dönük çalışmaların tümünü içerir. Veri Madenciliği uygulamaları aşağıdaki şekilde sınıflandırılabilirler. [5]

Bağıntı : “Gps alan müşterilerin %40 ‘ı bir arada satın alır.”

Kayıtlar incelenerek alanlar üzerindeki ilişkilerin tespiti için çalışılır. Bu çalışmadaki amaç alanlar arasındaki pozitif ve negatif korelasyonların tespiti edilmesidir. Örneğin meteorolojik bir veritabanındaki kayıtlar üzerinden yağın moramının %60 ve üzeri olduğu günlerin %55 ‘inde havanın yağışlı olduğunun tespiti edilmesi örnek olarak gösterilebilir. Burada amaç, zaten tahmin edilen gerçeğin gerçekleşme oranını hesaplamak değil, tahmin edilemeyen ilişkilerin ortaya çıkarılmasıdır.

Sınıflandırma: “Genç kadınlar küçük araba satın alırlar.”

Bir alanın özellikleri ile diğer bir alanın özelliklerinin eşleştirilmesi işlemidir. Bu işlemin satış bilgilerinin tutulduğu bir veritabanına uygulanması ile satılan ürünlerin ideal müşteri profili çıkarılabilir, ve bu bilgi reklamstratejilerinin oluşturulmasında temel yönlendirici olarak kullanılabilir.

Benzer Zaman Sıraları : “X şirketinin hisse senedi fiyatları ile Y şirketinin hisse senedi fiyatları benzer hareket ediyor.”

Bu analizle iki yada daha fazla hareket serisi arasında bağıntı kurulur. Bu bağıntı iki hastalığın toplum üzerinde dağılım konsantrasyonu olabilir. Dolayısı ile elde edilen ilişkiye dayalı olarak bir hastalığın görülme konsantrasyonunun artması ile bağıntılı hastalığın da sıklıkla görüleceği öngörülüp gerekli tedbirler alınabilecektir.

İstisnalar : “Şirket giriş çıkış saatleri çoğunlukta farklı olan çalışanlar hangileridir, hangi departmanda çalışıyorlar?”

Bu uygulamaya yapısında amaç öncekilerin aksine kural çıkarmaya yerine istisnai hareketleri tespitidir. Bir şirkete ait giriş çıkış kayıtlarından genel eğilime uymayan davranışlar tespit edilerek muhtemel yönetimsizlikler (iş dağılımının adaletsiz olması gibi) tespit edilebilir. Bu uygulamanın başka bir örneği olarak kredi kartı hareketlerinin analiz edilerek beklenmeyen alışverişlerin müşteri onayına bağlanması gösterilebilir.

Regresyon: “Ev sahibi olan, evli ve aynı işyerinde beş yıldan fazladır çalışan, geçmiş kredilerinde geç ödemesi bir ayı geçmemiş erkeğin kredi skoru 825 ‘tir”

Finans kurumlarında geçmiş kredi hareketlerinden elde edilen veriler ile oluşturulmuş bir kredi skor hesaplama algoritması geliştirilmiştir. Her kredi başvurusu için 0 ile 1000 arasında bir kredi skoru hesaplanır. Bu skor kişinin özellikleri ile geçmiş kredi hareketleri gözönüne alınarak hesaplanmıştır.

Zaman içinde Sıralı Örneklemler: “İlk üç taksiçiden iki yada üçünü geç ödeyen müşteriler %60 olasılıkla kanuni taksitleri giriyorlar.”

Burada amaç kredi almış kişilerin sonraki taksitlerini ödeme davranışlarını skorlamak amaçlanır. Dolayısıyla ödeme yönetiminde dikkat edilmesi gereken müşteri profili elde edilmiş olacaktır.

Doküman Madenciliği : “Arşivde belirli bir dokümana benzer dokümanlar hangileridir.”

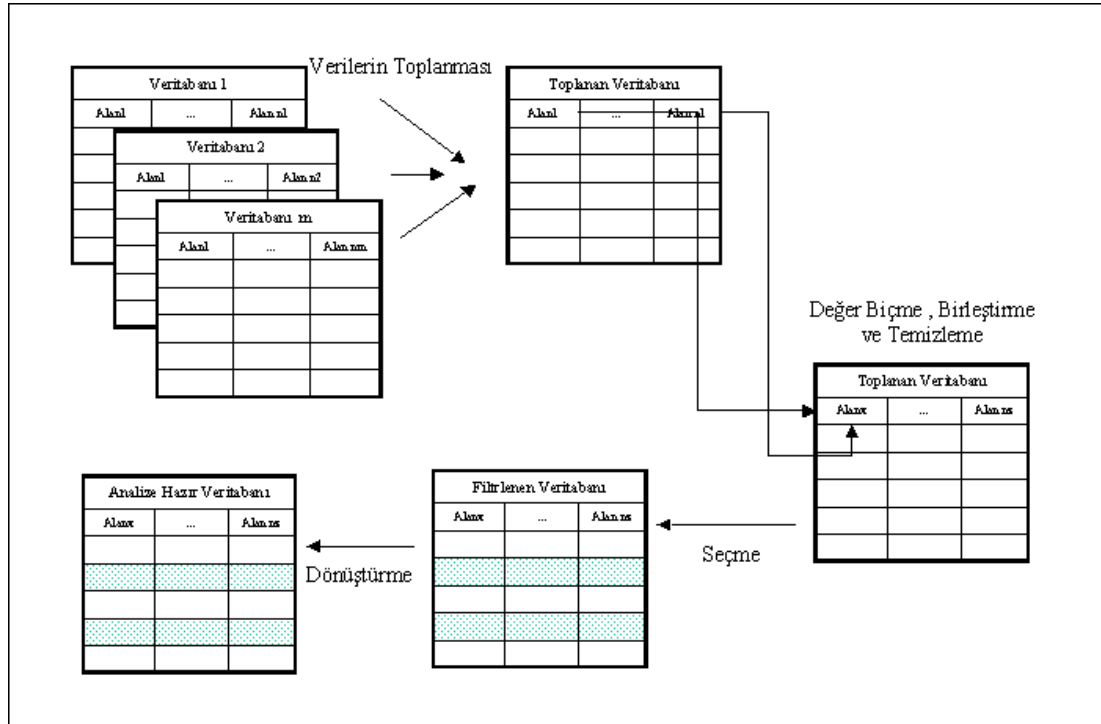
Amaç, otomatik olarak yapılan taraflar ile birbirine benzer dokümanların tespit edilmesi ve araştırmacılara kaynak tavsiyesinde kullanılmasıdır. Bu işlem genellikle önceden tespit edilen anahtar kelimeler aracılığı ile yapılmaktadır.

1.1.3 Bilgi Keşfi Süreci

Veri madenciliği çalışmalarında bilgi keşfi süreci beş ana adımda incelenebilir. [4] Bunlar,

- Problemin Tanınması
- Verilerin Hazırlanması
- Modelin Kurulması ve Değerlendirilmesi
- Modelin Kullanılması
- Modelin İzlenmesi

olarak listelenebilir. Ancak bu adımlardan önce aşılması gereken ilk aşama, üzerinde veri madenciliği çalışması yapılacak verideki tüm fonksiyonel ilişkilere detaylı haki mi yetin sağlanmasıdır. Bu analizi yapılacak sistemin iş akış yapısının ve ürettiği bilgini analistçe tam olarak anlaşılmasıdır. [4]



Şekil 1.2: Analizi için Kullanılacak Verinin Hazırlanması

Problemin Tanınması : Yapılacak veri madenciliği çalışmasında ilk geçilmesi gereken adım yapılacak çalışmanın sınırlarının tam olarak belirlenmesidir. Bu

aşamada analizi yapılacak sistemden ne tür bilgilerin süzülebileceği, maliyetinin ve işletmeye getirisinin ne olacağı açık bir şekilde izah edilmelidir.

Verilerin Hazırlanması : Veri madenciliği çalışmalarında analistin toplam çalışma süresinin en önemli bölümünü oluşturan fazdır. Bunun sebebi genellikle, veri analize tam olarak hazır değil iken analiz çalışması na başlanması, ve ara süreçlerde yaşanan geriye dönüşlerdir. Veri hazırlama kendi içerisinde toplama, değerlendirme, birleştirme ve temizleme, seçme ve dönüştürme adımlarından meydana gelmektedir.

Veri toplama adımı nda, verilerin çeşitli veri tabanı sistemlerinden toplanma işlemini yapar. Toplanan veriler genellikle birbirlerinden farklı yaklaşımlarla veri depolayan, birden fazla bağımsız sistemden bir araya getirilir. Değer biçme aşamasında toplanan verilerin uyumluluğu değerlendirilir, ve var olan uyumsuzlukların giderilme yolları araştırılır. Birleştirme ve temizleme aşamasında toplanan veriler aynı standartta birleştirilirler. Konsolide veri üzerinde var olabilecek uyumsuzluk ve anlamsızlıklar düzeltilir. Dördüncü aşama olan seçim sırasında elde edilen temiz veri grubu içerisinden analizi ilgilendiren alt grupların seçimi yapılır. Örneğin bir işletmenin sadece belirli bir tesisi üzerinde analiz yapılacak ise bu tesise ait veriler filtreleme ile ayrıştırılır. Son faz olan dönüştürmede, genellikle elde edilen veriler alt kısımlara bölünerek sınıflandırılırlar. Bu sayede daha anlamlı genel sonuçlara ulaşmak mümkün olur.

Modelin Kurulması ve Değerlendirilmesi : Tanımlanan problemi çözen en uygun modelin bulunabilmesi olabildiğince çok sayıda modelin kurularak denenmesi ile mümkündür. Bu nedenle veri hazırlama ve model kurma aşamaları, en iyi olduğu düşünülen modele varılıncaya kadar yinelenen bir süreçtir.[4]

Model kurulumu denetimli ve denetimsiz öğrenme şeklinde iki ayrı metodoloji ile yapılabilir. Denetimsiz öğrenmede örneklerin gözlenmesi ve alanlar arasındaki ilişkilere göre gruplamaların yapılması hedeflenir. Denetimli öğrenmede ise, modelin kurulumundan sonra bir grup veri analizi için, bir grup veri de analizin doğruluğunun kontrolü için ayrılarak modelin doğruluk derecesi elde edilir.

Modelin Kullanılması : Elde edilen model genellikle bir yazılım parçası olarak ortaya çıkar. Bu yazılım tek başına bir karar destek sistemi olarak tasarlanabileceği gibi mevcut yönetim sisteminin alt birimi olarak kullanılabilir.

Modelin İzlenmesi : Elde edilen tüm analiz yaklaşımlarının ürettikleri sonuçlar zaman içerisinde izlenmelidir. İşletmenin iş akışının değişmesi, çevre faktörlerinin etkilerinin farklılaşması gibi sebeplerle elde edilen analizler anlamını yitirmeye

başlayabilirler. Gereken durumlarda kullanılan analiz modeli revize edilerek değişen şartlara uyumu sağlanmalıdır.

1.1.4 Veri Madenciliğini Etkileyen Faktörler ve Uygulamanın Çetirileri

Veri madenciliği çalışmalarında beklenen sonuçlara ulaşabilmesi için gerekli olan üç gereklilik verinin yetirliliği, uzmanın deneyimi, ve araştırma çalışmalarına ayrılacak kaynak olarak karşımıza çıkmaktadır.

Sağlıklı analizlerin yapılabilmesi için, analizi yapılacak işletmenin olgunlaşmış veri işleme sistemine sahip olması ve iş akışı ve sonuçlarının ilgili veri saklama sistemlerinde düzenli şekilde tutuluyor olması gerekmektedir. Bu durumun sağlanmaması, hazine yetersiz ve sağlıklı veri ile yapılan analiz çalışmalarına yol açacak ve sonuçların güvenilirliğini olumsuz yönde etkileyecektir.

Daha önce değinildiği gibi verilerin anlamlı analizlerde kullanılabilmesi iş akışına hakim uzmanların desteği ile mümkün olacaktır. Dolayısıyla hedeflenen sonuçlara ulaşabilmek için gerekli uzman işgücünün, analiz çalışmalarına tahsis edilmesi gerekmektedir.

Veri madenciliği uzun zaman alan bir sürecin sonunda meyvelerini vermektedir. Bu süreç zaman maliyeti olduğu kadar yüksek analiz maliyetlerini de beraberinde getirir. İlgili maliyetler analiz projelerinin başında doğru hesaplanmalı ve göğüsleneyecek çalışmalar konusunda dikkatli olmalıdır.

Yukarıda ifade edilen maliyetler, çalışma doğru yapıldığında, yöneticilerine çok kıymetli destek bilgilerinin verilmesini sağlayacaktır. Günümüz rekabet ortamında bahsedilen kıymetteki bilgi işletmeyi verimlilik ve pazar stratejileri konusunda önemli kozlar ile donatacaktır.

1.2 Uzman Sistemler

Yapılacak veri madenciliği çalışmalarının sonuçları, çoğu zaman, uzman sistemler tarafından girildiği olarak kullanılabilirdiğinden uzman sistemler hakkında temel bilgilerin verilmesinde fayda vardır.

Uzmanlar belirli bir konuda bilgi birikimine sahip, karşılaşılan sorunların kaynağını tespit eden ve anlamlı çözümler üretebilen kişilerdir. Günlük yaşantımızda bilgi ve

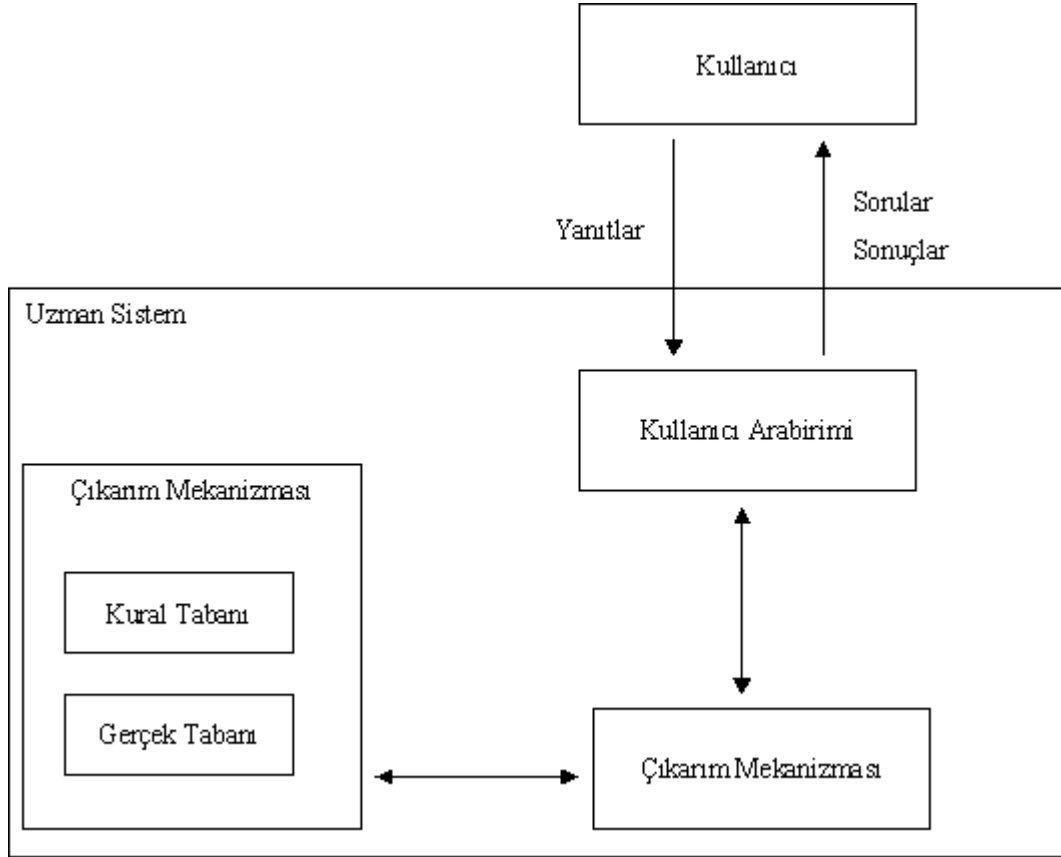
becerisine sık sık ihtiyaç duyduğumuz doktor, avukat, tamirci gibi mesleki gruplarının herbiri birer uzmanlık merkezidirler. Uzman sistemler, belirli bir alanda uzmanlaşmış personel yada çalışma gruplarının iş süreçlerini takip eden yazılım sistemlerinin geneldir. Uzman sistem tanıması, bir ülkenin savunma stratejilerini öneren dev bir sistemden, bir kelle işleminin yazım hatası bulan mekanizmasına kadar geniş bir yelpazeyi tanımlar.

1.2.1 Uzman Sistemin Bileşenleri

Uzman sistemler üç temel bileşenden oluşur. Bunlar,

- Bilgi Tabanı
- Çıkarım Mekanizması
- Kullanıcı Arabirimi

olarak listelenebilir. Şekil 1.3'te tipik uzman sistem bileşenleri gösterilmiştir.



Şekil 1.3: Uzman Sistemin Öğeleri

Bilgi tabanı uzman sistemin çıkarımları sırasında kullanılan gerçeklerin ve kurallarının saklandığı veri tabanı ortamıdır. Kural tabanı ve gerçek tabanı şeklinde iki alt öge olarak değerlendirilebilirler. Saklanan kurallar genellikle sistemin geliştirilmesi sürecinde uzmandan alınan ilişkilerdir. Kural yapıları “eğer ... ise...” şeklindedirler. Kurallar ağaç yapısında olabileceği gibi grift matris yapılar şeklinde de oluşturulabilirler. Gerçekler sistemin oluşturulması sırasında bilgi tabanına kaydedilebileceği gibi çoğunlukla, kullanımsırasında kullanıcıdan alınacak bilgiler ve kuralların sonuçları ile oluştururlar. Kural tabanları çıkarım mekanizmasından bağımsız çalışan birimlerdir. Dolayısıyla uzman sistemin kullanım döngüsü içerisinde, içerdiği bilgiler uzmanlar yada kullanıcılar tarafından değiştirilebilirler. Bu özellik bir uzman sistemi klasik yazılımlardan ayıran temel özelliklerden biridir. Bu sayede kullanıcı sadece bilgi tabanına yapacağı veri besleme ile programın akışı üzerine etki yapabilecek duruma gelir.

Çıkarım mekanizması kullanıcıdan alınan bilgiler ile kural tabanındaki kuralları tarayan ve yeni sonuçları bilgi tabanına yazan yazılım rutinleridir. Çıkarım mekanizmasının çalışması sırasında kullanıcıdan alınan her bilgi girişinde tetiklenir ve kural tabanındaki kurallar taranarak yeni gerçeklerin tespiti hedeflenir. Tespit edilen yeni gerçekler gerçek tabanına ilave edilir. Kuralları tetiklenmesi, kullanıcıya yeni soru sorulması döngüsü anlamı bir sonuca ulaşmaya yada sonuç çıkarılabilecek uygun kural kalıplarına kadar devam eder.

Kullanıcı arabirimi, çıkarım mekanizması ile kullanıcı arasındaki iletişimi sağlayan yazılım ortamıdır. Kullanıcı arabirimi doğrudan kullanıcı ile iletişimi kuran grafik bir ortam olabileceği gibi başka bir sistemle veri alışverişi sağlayan bir protokol olabilir.

Son zamanlarda ticari amaçla yazılan, uzman sistem kabukları denilen sistemler piyasaya sürülmektedir. Bu sistemler kendi içerilerinde çıkarım mekanizmaları ve bilgi içeriklerini bilgi tabanları içerirler. Bir uzman sistemin terzi usulü yazılımının maliyetinin yüksek bulunduğu durumlarda uzman sistem kabuklarından faydalananak tavsiye edilebilecek bir yöntem olabilir. Uzman sistem kabuğunun kullanımı için kabuğun bilgi tabanının işin uzmanı kişi tarafından doldurulması yeterli olmaktadır.

1.2.2 Uzman Sistem Kullanımının Faydaları

Uzman sistem kullanımı konuyla ilgili uzman istihdamının mümkün olduğu birçok durumda başvurulabilecek uygun bir çözümdür. Başlıca getirileri olarak

- Verimlilik artışı

- Maliyetin düşürülmesi
- Güvenilirlik
- Cevap verme süresinin kısalması
- Hizmetin devamlılığı

sayılabilir.

Uzman sistemlerin çıkarım hızlarında, artan bilgi ve iş yükü karşısında radikal değişimler olmaz. Uzman personel istidamı genellikle yüksek maliyetli ve riskli bir süreçtir. Uzman personelin belirli bir konuda motivasyonunun sağlanması ve işe bağlılıkla görev yapmasını beklemek çoğu zaman ulaşılmaz zor bir hedef iken, uzman sistemler için böyle bir risk ve yüksek maliyet söz konusu değildir.

Uzman sistemin ürettiği sonuçların bilgi tabanıyla uyumluluğu hemen hemen kesindir. Bunun nedeni, uzman sistemlerin, uzmanların aksine, eldeki tüm bilgi ve kuraları insani ihmallere ve duygusalılıklardan uzak bir şekilde değerlendirmesidir. Dar bilgi kümesi ve geniş zaman seçenekleri ile yüksek performanslı çalışan uzmanların çalışma süresinin daralması yada bilgisinin ani artışı durumlarında performanslarının düşmesi doğal bir durumdur. Uzman sistemler ise standart performanslarını değişen şartlarda koruyabilmekte ve önceden ölçeklenebilmekte dirler. Uzmanların yaşadıkları sağlık problemleri, iş değiştirme leri gibi hizmetin sürekliliğini aksatan problemlerin oluşması uzman sistem kullanımı ile aşılabılır.

2 BULANIK KÜMELER VE BULANIK MANTIK

Klasik Mantık tüm önermelerin iki değer ile değerlendirilmesi prensibi üzerine kurulmuştur. Yani her önermenin alabileceği sadece iki değer vardır; önermeler, insan düşünce sistemine ters bir şekilde, sadece iki değer ile notlandırılabilirler; doğru/yanlış. Bulanık Mantık, doğru/yanlış, evet/hayır, olumlu/olumsuz gibi iki değerli ilişkilendirmelerin ötesinde, üyelik ilişkilerinin tanımlanabildiği çok değerli bir mantık sistemidir. İnsani yaklaşımda, bir çok ilişkinin bu keskinlikte doğruluk değeri bulunmaz. Günlük yaşamda kullandığımız uzun, kısa, sıcak gibi grupların üyelerinin, evrensel kümenin hangi üyeleri olduğu net olarak belirlenmez. Bulanık mantık, klasik mantık ve klasik küme yaklaşımıyla aşılamayan bu sınırların matematiksel modellenmesi için kullanılan bir sistemdir.

Gelişiminin başından itibaren iki değerli mantık işlemlerinin hesaplama ve değerlendirilmesi için tasarlanan bilgisayarlar, bu kısıtlardan ötürü, insan gibi düşünen programları çalıştıramamışlardır. Bulanık mantığın gelişimi paralelinde, insan değerlendirilmesine yakın matematiksel modeller kurulabilir hale gelmektedir.

2.1 Bulanık Kümeler

Bulanık kümeler geleneksel küme tanımının genişletilmiş hali olarak değerlendirilebilirler.

Geleneksel Kümeler: Küme bütün olarak kabul edilebilecek nesnelere topluluğudur. Geleneksel küme teorisinin kurucusu Cantor'un tanımlamasına göre, kümenin elemanları kümeye aittir yada değildir. Günlük yaşamda bir grup olarak nitelendireceğimiz pek çok topluluk küme olarak değerlendirilebilir. Aşağıdaki örnekler geleneksel küme tanımına uyan gruplardır.

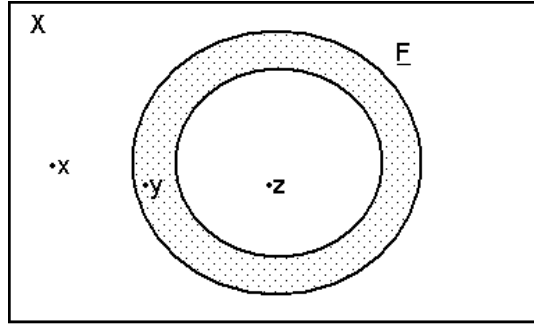
- (a) 1 ile 7 arasındaki çift sayılar : Bu tanımlama ile $\{2, 4, 6\}$ sayılarına ulaşılabildiğinden adı geçen topluluk küme tanıma uyar.

(b) Sağ olan Selçuklu Sultanları: Bu tanımlama üyesi olmayan bir kümeyi tanımlamaktadır. Zira, bir kişinin adı geçen kümenin üyesi olup olmadığı net olarak belirlenebilir.

(c) 5 ten küçük tam sayılar : Bu örnekteki topluluk sonsuz elemana sahip bir kümedir. Farkedileceği gibi bu örnekte de bir nesnenin kümenin üyesi olup olmadığına net olarak karar verilebilmektedir.

Örneklere de görüleceği gibi geleneksel kümeler üyeleri net olarak tespit edilebilen topluluklardır.

Bulanık Kümeler: Bulanık kümeler hatları net olarak ifade edilemeyen grupları tanımlamak için kullanılırlar. Genç insanlar, başarılı futbol takım gibi insanlar içinde sınırları tam olarak çizilemeyen gruplar üzerinde çalışmayı olanaklı hale getirirler. Bulanık Küme Teorisi ilk ifadesini 1960'lı yıllarda Zadeh ile bulmuştur. [6]



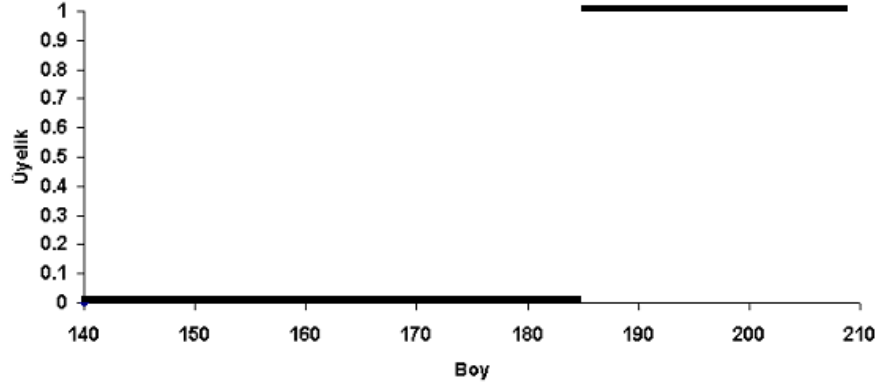
Şekil 2.1: \underline{F} Bulanık Kümesinin Diagram üzerindeki gösterimi

Şekil 2.1 'de, X evrensel kümesinde tanımlı \underline{F} bulanık kümesinin gösterimi yapılmıştır. Burada "x" bulanık kümenin elemanı değilken "z" ilgili bulanık kümenin tam elemanıdır. "y" 'nin ise \underline{F} bulanık kümesine tam olmayan üyeliği söz konusudur.

2.1.1 Karakteristik Fonksiyonun Özellikleri

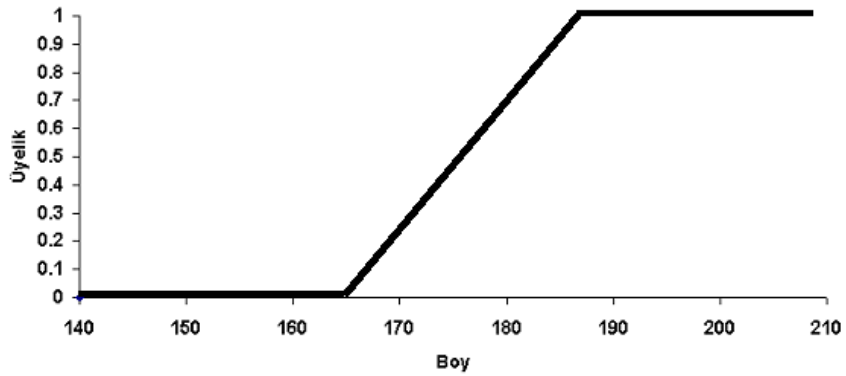
Bulanık kümeleri oluşturan elemanların herbirinin ilgili kümeye üyeliğini belirleyen bir üyelik değeri vardır. Üyelik derecesi $[0, 1]$ aralığında bir sayısal değer ile ifade edilir ve genellikle " μ " ile gösterilir. Üyelik değerinin 1 olması ilgili nesnenin kümenin tam üyesi olduğunu, 0 olması ise üyeliğin hiçbir şekilde bulunmadığını gösterir. Üyelik değerini belirleyen fonksiyonlara karakteristik fonksiyon yada üyelik fonksiyonu adı verilir. Geleneksel kümelerin karakteristik fonksiyonları $\{0, 1\}$ değerlerinden oluşan kesikli yapıya sahiptirler. Şekil 2.1 'de "x" elemanının \underline{F}

bulanık kümesine üyelik derecesi 0, “z” elemanının üyelik derecesi 1 ‘dir. “y” elemanın üyelik derecesi ise (0, 1) aralığında bir değere sahiptir.



Şekil 2.2 Geleneksel küme için karakteristik fonksiyon (uzun boylular kümesi)

“Uzun boylu insanlar” şeklinde ifade edilen bir bulanık kümeyi ele alalım 200cm boyunda olan bir şahsın bu kümenin elemanı olduğu aşikar bir şekilde görüldüğü gibi 145cm boyunda olan bir kişinin de kümenin elemanı olmadığı da açıkça bellidir. Ancak 175cm ile 185cm arasında boya sahip kişilerin bu kümeye ait olmaları kolayca tespit edilemez. Uzun boylu insanlar kümesi için geleneksel küme ve bulanık küme yaklaşımına göre karakteristik fonksiyonlar Şekil 2.2 ve Şekil 2.3 ‘de verilmiştir.



Şekil 2.3 Bulanık küme için karakteristik fonksiyon (uzun boylular kümesi)

Bulanık kümelere gösterimi evrensel kümenin sonlu ya da sonsuz elemanlı oluşuna göre iki şekilde yapılmaktadır. A bulanık kümesi için gösterim şekilleri (2.1) ve (2.2) 'de gösterilmiştir.

Evrensel Küme X sonlu sayı da elemana sahiptir;

$$A = \{ \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n \} = \sum_{i=1}^n \mu_A(x_i)/x_i \quad (2.1)$$

Evrensel Küme X sonsuz sayı da elemana sahiptir;

$$A = \int_{i=1}^{\infty} \mu_A(x_i)/x_i \quad (2.2)$$

Sonlu elemana sahip $X = \{ \text{uzunluk değerleri} \} = \{ 150, 160, 170, 180, 190, 200, 210 \}$ evrensel kümesini ele alalım. X üzerinde “Kısa Boylu”, “Orta Boylu”, “Uzun Boylu” bulanık kümelerini tanımlayacak olursak Tablo 2.1 'deki gibi bir dağılım elde ederiz.

Tablo 2.1: X de tanımlı bulanık kümeler ve üyelik değerleri

Uzunluklar (cm)	Kısa Boylu	Orta Boylu	Uzun Boylu
150	1	0,3	0
160	0,7	0,4	0
170	0,3	1	0
180	0	1	0,4
190	0	0,3	0,7
200	0	0	1
210	0	0	1

Tablo 2.1 'de tanımlı bulanık kümeler,

$$\text{Kısa_Boylular} = \{ 150/1 + 160/0,7 + 170/0,3 + 180/0 + 190/0 + 200/0 + 210/0 \}$$

$$\text{Orta_Boylular} = \{ 150/0,3 + 160/0,4 + 170/1 + 180/1 + 190/0,3 + 200/0 + 210/0 \}$$

$$\text{Uzun_Boylular} = \{ 150/0 + 160/0 + 170/0 + 180/0,4 + 190/0,7 + 200/1 + 210/1 \}$$

şeklinde gösterilirler.

2.1.2 Bulanık Kümelerin Özellikleri

Bulanık kümeler üzerinde tanımlanan özelliklerin başlıcaları aşağıda ifade edilmiştir.[3]

Çekirdek (core): Evrensel kümesini \underline{A} bulanık kümesine tam olarak üye olan elemanları kümesidir. Çekirdekteki elemanların bulanık kümeye üyelik dereceleri 1 'dir.

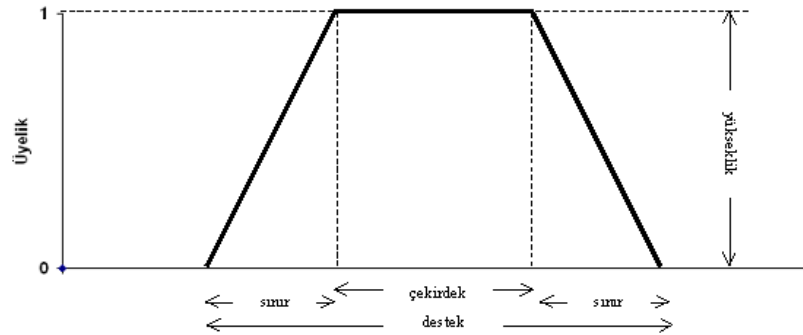
$$core(\underline{A}) = \{x \in X \mid \mu_A(x) = 1\} \quad (2.3)$$

Sınır (boundry): Evrensel kümesini \underline{A} bulanık kümesine üye olan ancak üyelik değeri 1 'den küçük olan elemanları kümesidir. Sınırı boş olan bulanık kümeler belirgin küme olarak sınıflandırılırlar.

$$boundary(\underline{A}) = \{x \in X \mid 0 < \mu_A(x) < 1\} \quad (2.4)$$

Destek (support): Evrensel kümesini \underline{A} bulanık kümesine üye olan elemanları kümesidir. Başka bir deyişle bulanık kümenin çekirdek ve sınırının birleşimidir.

$$core(\underline{A}) = \{x \in X \mid \mu_A(x) > 0\} \quad (2.5)$$



Şekil 2.4 Bulanık kümenin bazı temel özellikleri

Yükseklik (core): \underline{A} bulanık kümesine üye olan elemanların en yüksek üyelik değeri dir.

$$height(\underline{A}) = \max(\mu_A(x)) \quad (2.6)$$

height(\underline{A}) = 1 olan bulanık kümeler normal küme olarak adlandırılırlar. Farkedileceği gibi normal bir kümenin en az bir elemanı kümeye tam olarak üyedir.

Kardinalite (cardinality): \underline{A} bulanık kümesine üye olan elemanların üyelik değerleri toplamıdır. $|\underline{A}|$ ile gösterilir.

$$|\underline{A}| = \sum \mu_A(x) \quad (2.7)$$

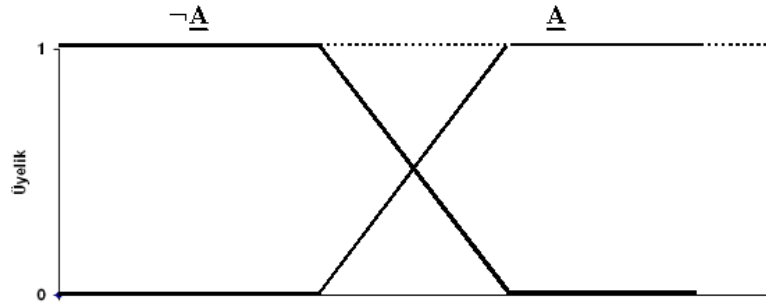
2.1.3 Temel Bulanık Küme İşlemleri

Bulanık kümeler ile ilgili tanımlanan birleşme, kesişme, tümlenen işlemlerinin tanımları aşağıdaki gibidir. [3]

Tümlenen Küme: \underline{A} bulanık kümesinin tümleneni $\neg A$ ile gösterilir, $\mu_A(x)$, x elemanının A kümesine üyelik derecesi olmak üzere aşağıdaki gibi tanımlanır.

$$\mu_{\neg A}(x) = 1 - \mu_A(x) \quad (2.8)$$

Şekil 2.5 'te bulanık tümlenen küme için üyelik fonksiyonu görülmektedir.



Şekil 2.5 Bulanık kümenin değili

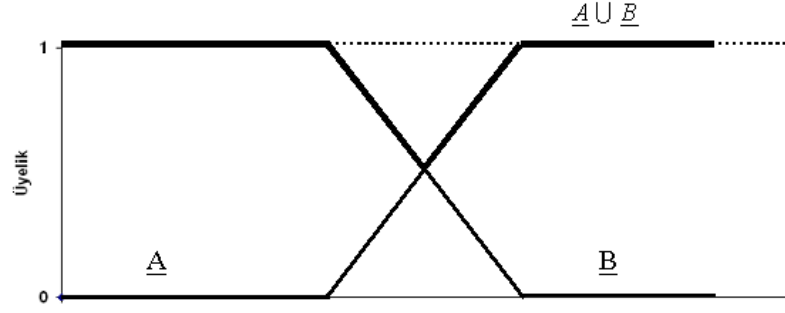
Tablo 2.1 'deki dağılım göz önüne alırsak "Kısa Boylular" kümesinin tümleneni aşağıdaki gibi elde edilir.

$$\neg \text{Kısa_Boylular} = \{ 150/0 + 160/0,3 + 170/0,7 + 180/1 + 190/1 + 200/1 + 210/1 \}$$

Bulanık Birleşme: \underline{A} ve \underline{B} X de tanımlı iki bulanık küme ve sırasıyla $\mu_A(x)$ ve $\mu_B(x)$ bu kümelerin üyelik fonksiyonları olsun. Bu durumda, $\underline{A} \cup \underline{B}$ 'de bir bulanık kümedir ve bir x elemanın $\underline{A} \cup \underline{B}$ kümesine üyelik derecesi $\mu_{\underline{A} \cup \underline{B}}(x)$ ile gösterilir.

$$\mu_{\underline{A} \cup \underline{B}}(x) = \max(\mu_A(x), \mu_B(x)) \quad (2.8)$$

Şekil 2.6 'da bulanık birleşme için üyelik fonksiyonu gösterilmektedir.



Şekil 2.6: Bulanık birleşme için üyelik fonksiyonu

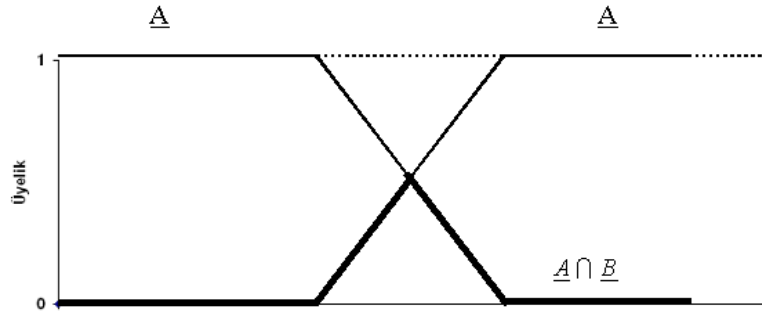
Tablo 2.1 'deki dağılım göz önüne alırsak "Kısa_Boylular Y Uzun_Boylular =K Y U" kümesi aşağıdaki gibi elde edilir.

$$K Y U = \{150/1 + 160/0,7 + 170/0,3 + 180/0,4 + 190/0,7 + 200/1 + 210/1\}$$

Bulanık Kesişme: \underline{A} ve \underline{B} X de tanımlı iki bulanık küme ve sırasıyla $\mu_{\underline{A}}(x)$ ve $\mu_{\underline{B}}(x)$ bu kümelerin üyelik fonksiyonları olsun. Bu durumda, $\underline{A} \cap \underline{B}$ 'de bir bulanık kümedir ve bir x elemanın $\underline{A} \cap \underline{B}$ kümesine üyelik derecesi $\mu_{\underline{A} \cap \underline{B}}(x)$ ile gösterilir.

$$\mu_{\underline{A} \cap \underline{B}}(x) = \min(\mu_{\underline{A}}(x), \mu_{\underline{B}}(x)) \quad (2.8)$$

Şekil 2.7 'te bulanık kesişme için üyelik fonksiyonu görülmektedir.



Şekil 2.7: Bulanık kesişme için üyelik fonksiyonu

Tablo 2.1 'deki dağılım göz önüne alırsak "Kısa_Boylular Y Uzun_Boylular = K I U" kümesi aşağıdaki gibi elde edilir.

$$K I U = \{150/0 + 160/0 + 170/0 + 180/0 + 190/0 + 200/0 + 210/0\}$$

Bul an k Alt Kü me: \underline{A} ve \underline{B} X de tam nhl i ki bul an k kü me ve sırasıyla $\mu_{\underline{A}}(x)$ ve $\mu_{\underline{B}}(x)$ bu kü mel eri n üyel ik fonksi yonları olsun

$$\forall x \in X, \mu_{\underline{A}}(x) < \mu_{\underline{B}}(x) \quad (2.9)$$

şartı sağ lanı yorsa $\underline{A} \subseteq \underline{B}$ 'nin bul an k alt kü mesi dir denir.

2.1.4 Bul an k Kü mel eri n Özelli kleri

Geleneksel kü mel er için geçerli olan özelli kleri n bir çoğu bul an k kü mel er için de geçerli dir. Temel bul an k kü me özelli kleri aşağı da listelenmiştir.[3]

a) Birleş me $\underline{A} \cup (\underline{B} \cap \underline{C}) = (\underline{A} \cup \underline{B}) \cap \underline{C}$ (2.10)

$$\underline{A} \cap (\underline{B} \cup \underline{C}) = (\underline{A} \cap \underline{B}) \cup \underline{C} \quad (2.11)$$

b) Değiş me $\underline{A} \cup \underline{B} = \underline{B} \cup \underline{A}$ (2.12)

$$\underline{A} \cap \underline{B} = \underline{B} \cap \underline{A} \quad (2.13)$$

c) Dağı l ma $\underline{A} \cup (\underline{B} \cap \underline{C}) = (\underline{A} \cup \underline{B}) \cap (\underline{A} \cup \underline{C})$ (2.14)

$$\underline{A} \cap (\underline{B} \cup \underline{C}) = (\underline{A} \cap \underline{B}) \cup (\underline{A} \cap \underline{C}) \quad (2.15)$$

d) De Morgan $\neg(\underline{A} \cup \underline{B}) = \neg \underline{A} \cap \neg \underline{B}$ (2.16)

Kuralı

$$\neg(\underline{A} \cap \underline{B}) = \neg \underline{A} \cup \neg \underline{B} \quad (2.17)$$

e) Geçiş me $\underline{A} \subseteq \underline{B}$ ve $\underline{B} \subseteq \underline{C}$ ise $\underline{A} \subseteq \underline{C}$ (2.18)

2.2 Bul an k Mantık

Klasik mantık öner mel eri n doğru ve yanlış şeklinde iki değerle değerlendirilebileceği temeli üzerine kurul müştür. Bu kısıtlı na bir çok alanda klasik mantığın kullanı mını engellemiş, bilim dünyasını yeni oluşumlara yönl endir müştür. Zaman içerisinde iki değerli klasik mantığın üç değerli mantığa genişlet me çalış maları ol müştür. Ortaya çıkan çeşitli modeller, kendi içerisinde kurallara sahip ol müşler dir. He men he men tüm çalış malar da doğruluk durumu 1, yanlışlık durumu 0 ve belirsizlik durumu 0.5 ile gösteril müştür. O uşturulan üç değerli mantık modelleri ni n hiç biri klasik mantığın temeli ni ol uşturan aksiyom ve totolojilerin hepsini birden sağ laya nam müştür. Bu

yetersizlik üç değerli mantık sistemlerinin etkili kullanımını ve yaygınlaşmasına engel olmuştur.[6]

Bulanık mantık, bulanık küme teorisini temel dayanak noktası olarak Zadeh tarafından 1960'lı yılların ortasında ortaya atılmıştır. Zadeh 'in kurduğu model, bulanık kümelerin üyelik değerlerine benzer bir şekilde, önermelerin $[0, 1]$ aralığında doğruluk değerleri alınması prensibi ne dayanmaktadır.

Bulanık mantık ile doğal konuşma dilinde var olan çok az, genellikle gibi belirsiz ifadelerin matematisel model oluşturulması, doğruluğu tam olarak notlandırılmayan önermelerin işlenebilmesi mümkün olmuştur.

2.2.1 Doğruluk Değeri

Günlük yaşamda kullanılan ifadelerin doğruluğu çok az durumda kesin olarak ifade edilebilir. İfade edilen önermeler kişiden kişiye değişen anlamlar taşırlar. Dolayısıyla çevremizdeki tüm sistemimiz mi çin belirsizdir. 'Su sıcaktır.' şeklindeki bir önermenin doğruluk değeri kişiler tarafından farklı değerlendirilecektir. Bu söylemlerin pek çoğu birer bulanık önermedir.

Her \underline{P} bulanık önermesinin $[0, 1]$ arasında birer doğruluk değeri vardır ve $T(\underline{P})$ ile gösterilir. Bulanık kümelerin doğruluk değerlerini, bulanık kümelerin üyelik değerlerine karşılık getirebiliriz

Örneğin, \underline{X} ve \underline{A} kümeleri

$$\underline{X} = \{\text{Su Sıcaklıkları}\} = \{0^{\circ}, 20^{\circ}, 40^{\circ}, 60^{\circ}, 80^{\circ}, 100^{\circ}\}$$

$$\underline{A} = \{\text{Sıcak Su}\} = [0/0^{\circ}, 0/20^{\circ}, 0.2/40^{\circ}, 0.5/60^{\circ}, 0.8/80^{\circ}, 1/100^{\circ}]$$

şeklinde tanımlanmış olsun. Bu durumda 80° sıcaklığındaki bir sürahi su için yapılacak $\underline{P} = \text{"Sürahi deki su sıcaktır."}$ önermesinin doğruluk değeri $T(\underline{P}) = 0.7$ olacaktır.

2.2.2 Temel Bulanık Mantık İşlemleri

\underline{P} ve \underline{Q} bulanık önermeler, $T()$ doğruluk fonksiyonu olmak üzere temel mantıksal işlemlerin doğruluk değeri aşağıdaki gibi tanımlanır.

a) $\underline{P} \vee \underline{Q}$ (\underline{P} veya \underline{Q}) :

$$T(\underline{P} \vee \underline{Q}) = \max(T(\underline{P}), T(\underline{Q})) \quad (2.19)$$

b) $\underline{P} \wedge \underline{Q}$ (P ve Q) :

$$\mathbb{T}(\underline{P} \wedge \underline{Q}) = \min(\mathbb{T}(\underline{P}), \mathbb{T}(\underline{Q})) \quad (2.20)$$

c) $\neg \underline{P}$ (değil P) :

$$\mathbb{T}(\neg \underline{P}) = 1 - \mathbb{T}(\underline{P}) \quad (2.21)$$

d) $\underline{P} \rightarrow \underline{Q}$ (P gerektirir Q) :

$$\mathbb{T}(\underline{P} \rightarrow \underline{Q}) = \max(\mathbb{T}(\neg \underline{P}), \mathbb{T}(\underline{Q})) \quad (2.22)$$

2.2.3 Dilsel Pekiştiriciler

Kullanılan doğal dilde “çok”, ”az”, ”çok az” gibi eklemeler ile önermeler üzerinde güçlendirici yada zayıflatıcı vurgular yapılır. Dilsel pekiştiriciler anlam vurguladıkları gibi mevcut bulanık önerme ve kümelerden yeni bulanık önerme ve kümelerin türetilmesinde sebep olurlar. Örneğin uzun boylular bulanık kümesi üzerinde kullanılacak “çok” yada “çok çok” pekiştiricileri ile “çok uzun boylular” ve “çok uzun boylular” gibi iki bulanık küme daha elde etmiş oluruz. [6]

Doğal dil ile söylenen pekiştirici ifadeleri bulanık kümeler aracılığı ile matematiksel olarak ifade edebiliriz. Pekiştirici ifadenin bulanık küme üzerine uyarlanması üyelik fonksiyonunun bir pekiştirici fonksiyonla işlem sokulması ile olur. Dilsel pekiştiricilerin modele uygulanması ile doğruluk değerleri üzerine azaltıcı yada arttıcı etki yapılmış olur. Kullanılan matematiksel fonksiyonun seçiminde doğal dil ile ifade edilme istenen vurgu ile, doğruluk değeri arasındaki ilişkinin korunması na dikkat edilmelidir.

Aşağıda bazı temel dilsel pekiştiriciler verilmiştir. [6]

a) Çok :

$$\underline{A}^2 = \sum_{i=1}^n \mu_A^2(x_i) / x_i \quad (2.23)$$

b) Aşırı :

$$\underline{A}^3 = \sum_{i=1}^n \mu_A^3(x_i) / x_i \quad (2.23)$$

c) Çok Çok

$$\underline{A}^4 = \sum_{i=1}^n \mu_A^4(x_i) / x_i \quad (2.24)$$

c) Yaklaşık

$$\underline{A}^{0.5} = \sum_{i=1}^n \mu_A^{0.5}(x_i) / x_i \quad (2.25)$$

3. VERİ TABANLARI NDAKİ BULANIK BAĞLI LI KLAR

İlişkisel veri tabanı modelinin ortaya çıkmasıyla veri tabanı sistemleri daha işlevsel hale geldiler. Tutulan verilerin düzenli saklanması sayesinde, kayıtlardaki gereksiz tekrarlar önendi, raporlamalarda önemli imkanlar sağlandı. İlişkisel veri tabanı modelinden önce değişik veri tabanı tasarımları yapılmış olsa da, bunlar ilişkisel veri tabanı modelinin yaygınlığına ve uygulanabilirliğine ulaşamadılar.

İlişkisel veri tabanlarında tanımlanan kısıtlar ve kurallar ile verilerin tutarlılığı sağlanmaktadır. Bütünsel kısıtlamalar (integrity constraints-IC) sistem üzerinde genel kurallar oluşturulabilmektedir. Bütünsel kısıtlamalar, genellikle, uygulamaya seviyesinde yazılan, takip edilen sürecin mantıksal bütünlüğünü korumaya amaçlı üretilen kontrol yazımlarıdır. Basit, bir örnek olarak “bir öğrenci aynı ders saatinde en fazla bir derse girebilir” verilebilir.[1]

Veri tabanlarının kullanımı sürecinde, toplanan veri üzerinde inceleme yapan analistler iki önemli kavramla yüzleşmişlerdir; fonksiyonel bağımlılık(functional dependency(FD)), dereceli bağımlılık(gradual dependency(GD))

Bu yüksek lisans tezinde fonksiyonel ve dereceli bağımlılıkları hesaplayan bir sistemin geliştirilmesi hedeflenmiş ve Dan Rasmussen ile Ronald R. Yager tarafından 1996 tarihinde yayınlanan algoritma kullanılmıştır.

3.1 Bulanık Fonksiyonel Bağımlılık ve Dereceli Bulanık Bağımlılık

Bulanık fonksiyonel bağımlılık(FFD) veri tabanındaki alanların birbirleriyle birebir ilişkili olmaları durumunu gösteren bir parametredir. FFD için, “Belirli bir bölgede yaşayanlar, belirli bir ürünü kullanırlar.” örnek olarak verilebilir. Dikkat edileceği gibi burada kuralı oluşturan iki önerme, veri tabanındaki iki ayrı alandan gelmekte ve benzerlik ilişkisi aranmaktadır. [1]

Dereceli bulanık bağımlılık(GFD) ise, veri tabanındaki alanların taşıdıkları değerlerin paralel olmaları durumunu gösteren parametredir. GFD’ye örnek olarak, “Maaşı çok olanlar, çok kredi kartı borcu öderler” gösterilebilir. Burada, veri

tabanındaki farklı alanların sayısal olarak aynı yönde hareket edip etmediği incelenir.

Tezde uygulanan algoritma ile Oracle tabanlı bir veri tabanındaki herhangi bir tablodaki GFD ve FFD ilişkiler sögulanabilmekte ve ilişkisinin doğruluğunu gösteren bir sayısal doğruluk değeri elde edilebilmektedir.

3.2 Dİsel Özetleme

Hesaplama algoritmalarının açıklanmasından önce dİsel özet cümlelerini genel for munda bahsetmekte yarar vardır. Dİsel özet cümleleri çeşitli for mlarda olabilir. Tez çalışmasında kullanılan for maşığı da verildiği gibidir: [1]

DB veritabanındaki **R** özelliğindeki kayıtların **Q** kadarını aynı zamanda **S** özelliğindedir.

Örneğ verecek olursak

Fatura veritabanı 'ndaki **20 kalemin üzerindeki** faturaların **çoğu bayanlara** aittir.

Burada S'ye özetleyici, R'ye alt topluluk ve Q'ya ise özetteki nicelik adı verilir. Örneğimizdeki öğeleri eşleştirecek olursak, "20 kalemin üzerinde kaleme sahip faturalar" R, "bayanlara ait faturalar" S, "çoğu" Q olarak elde edilir.

Her özetleme cümlesine $T \in [0, 1]$ şeklinde bir doğruluk değeri atanır. Doğruluk değeri ilgili özet cümlesinin veri tabanındaki kayıtlara uygulanabilirliğini sayısal olarak temsil eder. Özet cümlesinin yapı taşlarından özetleyici S ve alt topluluk R genellikle bulun k kümelere olmakla birlikte, klasik kümelere özet işlemlerinde özetleyici ve alt topluluk olarak kullanılabilir.

Cümleciğe atanan doğruluk değeri T'nun hesaplanmasıyla ilgili Yager ve Rasmussen'in önerdikleri algoritma aşağıda detaylı olarak verilmiştir.

DB veritabanının adet nesneye sahip bir grup olarak ele alalım ($DB = \{o_1, o_2, K, o_n\}$)

Uygulamada n adet kayda sahip bir ilişkisel veri tabanı tablosu DB'nin yerine kullanılabilir. S ve Q DB üzerinde tanımlı bulun k kümelere olsun

1. Her bir o_i nesnesinin R ve S bulun k kümelere üyeli k değerleri olan $R(o_i)$ ve $S(o_i)$ hesaplanır.

2. S bul an k kümesi ne üye olan nesnel eri n R bul an k kümesi ne üye ol an lar ını n n oran ını gösteren φ hesap lan ır.

$$\varphi = \frac{\sum_{i=1}^n T(S(o_i), R(o_i))}{\sum_{i=1}^n R(o_i)} \quad (3.1)$$

3. Son ol arak ta, $T=Q(\varphi)$ ile T^* nun değeri hesap lan mış ol ur.

φ ' ni n hesap lan mas ında kullan ılan $T()$ fonksiyonu bir t-nor m fonksi yonudur. Uygul a ma lar da genelli kle mi ni mu m fonksi yonu ol arak al ın ır.

Özel dur um ol arak alt topl ul uğun tüm veri ta ba nı ol mas ını in cel eyeli m Eđer R alt topl ul uğu tüm veri ta ba nı ol arak al ın ır sa a şa ğı da ki for mül ü el de ede riz.

$$\varphi = \frac{\sum_{i=1}^n T(S(o_i), DB(o_i))}{\sum_{i=1}^n DB(o_i)} \quad (3.2)$$

Gör üle ce ği gi bi tüm o_i 'leri n DB topl ul u ğuna üye li ği ta m dır. Ya ni DB bul an k kü me de ğil dir ve tüm o_i 'ler i çin $DB(o_i)=1$ el de edil ir. Bu ra dan,

$$\varphi = \frac{\sum_{i=1}^n T(S(o_i), 1)}{\sum_{i=1}^n 1} \quad (3.3)$$

T fonksi yonu ol arak mi ni mu ma lı ndı ğı nda her $o_i \in DB$ i çin $S(o_i) < 1$ ol ur. Ya ni $T(S(o_i), 1) = S(o_i)$ el de edil ir. Bu ra dan da,

$$\varphi = \frac{\sum_{i=1}^n S(o_i)}{n} \quad (3.4)$$

el de edil ir.

T' 'yu hesap la mak i çin ol arak hesap lan an φ değeri $Q()$ fonksi yonuna ile iş le ne sok ul ur. Q az , çok , yak la şık gi bi d il sel vur gul ay ıcı lar ını özet cüm le ci ği ni n do ğ ru luk değeri ne yans ıtan o pe ra t ö r d ü r. D il sel vur gul uy u cu lar a 2.2.3 'te de ği nil mi ş tir. φ değeri ni $Q()$ 'fonksi yonu ile iş le ne sok tu ğu mu z da ,

$$\mathcal{T} = \mathbf{Q} \left(\frac{\sum_{i=1}^{\infty} S(o_i)}{n} \right) \quad (3.5)$$

elde ederiz.

3.3 Bulanık Fonksiyonel Bağımlılıklar (FFD)

Veri tabanındaki bir tablonun alanların birbirine bağlı hareketi fonksiyonel bağımlılık olarak adlandırılır. X alanı ile, Y alanı arasındaki bulanık olmayan fonksiyonel bağımlılık $X \rightarrow Y$ şeklinde gösterilir, ve o_i 'nin aşağıda ifade edilen şartı sağlama durumu fonksiyonel bağımlılığın varlığından bahsedilir.

$$\forall o_i, o_j \mid o_i.X = o_j.X \Rightarrow o_i.Y = o_j.Y \quad (3.6)$$

Bulanık fonksiyonel bağımlılık (fuzzy functional dependency (FFD)), yukarıda tanımlanan fonksiyonel bağımlılığın bulanık yapıya genelleştirilmesiyle elde edilir. X, Y bulanık kümelere arasındaki bulanık bağımlılık $X \rightarrow_F Y$ şeklinde gösterilir. Tüm bulanık yaklaşımlarda olduğu gibi, bulanık fonksiyonel bağımlılığa bir $\mathcal{T} \in [0, 1]$ doğruluk değeri ile eşleştirilir. \mathcal{T} 'nin 0 'a yakın olması bulanık bağımlılığın zayıf, 1 'e yakın olması ise güçlü olduğunu gösterir gösterir. $\mathcal{T} = 1$ durumu ilgili alanlar arasında normal fonksiyonel bağımlılığı gösterir. [1]

Veri tabanındaki alanlar arasındaki FFD 'nin \mathcal{T} doğruluk değeri hesaplamak için veritabanındaki her o_k elemanı ile aşağıdaki yapıda kurallar oluşturulur.

$$R_k : \text{eğer } o_k.X \approx o_j.X \text{ ise } o_k.Y \approx o_j.Y$$

$$R_k : \text{if } o_k.X \approx o_j.X \text{ then } o_k.Y \approx o_j.Y$$

Elde edilen her R_k kuralı için bir T_k doğruluk değeri hesaplanır. Bunun için aşağıdaki eşitlik kullanılır.

$$T_k = \sum_{Id} (o_i \in \{ o_j \in DB \mid o_k.X \approx o_j.X \} \mid o_k.X \approx o_i.X) \quad (3.7)$$

Burada kullanılan Id dilsel özetleyici birim belirleyicisi olup $Id(x) = x$ alınacaktır. $X \rightarrow_F Y$ FFD 'sinin doğruluk değeri \mathcal{T} , T_k değerlerinin ortalaması alınmasıyla elde edilir. Hesaplama işlemi sırasında benzerlik fonksiyonun (\approx) seçimi büyük önem arz etmektedir. Uygun seçimden yansıtılan benzerlik fonksiyonu gerçeği yansıtan

fonksiyonel bağımlılıkların çıkarılmasına yol açabilir. Benzerlik fonksiyon, analiz edilen veri sisteminin uzmanları tarafından belirlenmelidir.

Tablo 3.1 'de verilen örnek veritabanındaki Alan1 ve Alan2 arasındaki FFD 'nin doğruluk değerini hesaplayalım

Tablo 3.1 : FFD hesaplaması için örnek veritabanı

	Alan1	Alan2
o_1	1	1
o_2	2	4
o_3	4	6
o_4	8	6

Alan1 ve Alan2 arasındaki FFD 'nin doğruluğunun hesaplanması için tüm nesnelere ilişkin alanları arasındaki benzerliklerin doğruluk değerleri hesaplanmalıdır. Hesaplama işlemi sırasında benzerlik için doğruluk fonksiyonu olarak $\mu(x \approx y) = \max((4 - |x - y|)/4, 0)$ seçilmiştir. Hesaplamalar yapıldığında aşağıdaki liste elde edilir.

Tablo 3.2: FFD hesaplaması için gerekli ara hesaplamalar

o_i	$\mu(o_i, \text{Alan1} \approx o_j, \text{Alan1})$	$\mu(o_i, \text{Alan2} \approx o_j, \text{Alan2})$
o_1	$\mu(1 \approx 1) = 1$	$\mu(1 \approx 1) = 1$
	$\mu(1 \approx 2) = 0.75$	$\mu(1 \approx 4) = 0.25$
	$\mu(1 \approx 4) = 0.25$	$\mu(1 \approx 6) = 0$
	$\mu(1 \approx 8) = 0$	$\mu(1 \approx 6) = 0$
o_2	$\mu(2 \approx 1) = 0.75$	$\mu(4 \approx 1) = 0.25$
	$\mu(2 \approx 2) = 1$	$\mu(4 \approx 4) = 1$
	$\mu(2 \approx 4) = 0.5$	$\mu(4 \approx 6) = 0.5$
	$\mu(2 \approx 8) = 0$	$\mu(4 \approx 6) = 0.5$
o_3	$\mu(4 \approx 1) = 0.25$	$\mu(6 \approx 1) = 0$
	$\mu(4 \approx 2) = 0.5$	$\mu(6 \approx 4) = 0.5$
	$\mu(4 \approx 4) = 1$	$\mu(6 \approx 6) = 1$

	$\mu(4 \approx 8) = 0$	$\mu(6 \approx 6) = 1$
o_4	$\mu(8 \approx 1) = 0$	$\mu(6 \approx 1) = 0$
	$\mu(8 \approx 2) = 0$	$\mu(6 \approx 4) = 0.5$
	$\mu(8 \approx 4) = 0$	$\mu(6 \approx 6) = 1$
	$\mu(8 \approx 8) = 1$	$\mu(1 \approx 6) = 1$

Veritabanındaki kuralları çıkarırsak,

- R_1 : eğer $1 \approx o_i$. Alan1 ise $1 \approx o_i$. Alan2
 R_2 : eğer $2 \approx o_i$. Alan1 ise $4 \approx o_i$. Alan2
 R_3 : eğer $4 \approx o_i$. Alan1 ise $6 \approx o_i$. Alan2
 R_4 : eğer $8 \approx o_i$. Alan1 ise $8 \approx o_i$. Alan2

kuralları elde edilir. T_i doğruluk değerlerini elde etmek için

$$T_i = \frac{\sum_{k=1}^n T(\mu(o_i.x \approx o_k.x), \mu(o_i.y \approx o_k.y))}{\sum_{k=1}^n \mu(o_i.x \approx o_k.x)} \quad (3.8)$$

formülünde T fonksiyonunu minimum olarak seçersek ve örneğimizi göre düzenlersek,

$$T_i = \frac{\sum_{k=1}^n \min(\mu(o_i.Alan1 \approx o_k.Alan1), \mu(o_i.Alan2 \approx o_k.Alan2))}{\sum_{k=1}^n \mu(o_i.Alan1 \approx o_k.Alan2)} \quad (3.9)$$

elde ederiz. Hesaplamalar sonunda R_i kuralları için doğruluk değerleri aşağıdaki gibi elde edilir.

$$T_1 = \frac{1+0.25+0+0}{1+0.75+0.25+0} = 0.625$$

$$T_2 = \frac{0.25+1+0.5+0}{1+0.75+0.5+0} = 0.89$$

$$T_3 = \frac{0+0.5+1+0}{0.25+0.5+1+0} = 0.89$$

$$T_4 = \frac{0+0+0+1}{0+0+0+1} = 1$$

Son olarak $\{Alan1\} \rightarrow_F \{Alan2\}$ için doğruluk değeri,

$$\mathcal{T} = \frac{T_1 + T_2 + T_3 + T_4}{4} = \frac{0.625 + 0.89 + 0.86 + 1}{4} = 0.84$$

şeklinde hesaplanır. Benzer bir şekilde, $\{Alan2\} \rightarrow_F \{Alan1\}$ için doğruluk değeri de hesaplanabilir.

3.4 Dereceli Bulanık Bağımlılıklar (GFD)

Dereceli bulanık bağımlılık (GFD) veritabanındaki alanlar arasındaki yönlilişkilerinin tespiti için kullanılan bir ölçüdür. Genel formu

veritabanındaki G_1 nesnelere G_2 olanlardır

şeklinde dir. Fonksiyonel bulanık bağımlılık kurallarında olduğu gibi dereceli bulanık bağımlılık kurallarına da birer $\mathcal{T} \in [0, 1]$ doğruluk değeri karşılık getirilir. Doğruluk değerinin 0'a yakın olması dereceli bulanık bağımlılığın zayıf, 1'e yakın olması ise güçlü olduğunu gösterir. Dereceli bulanık bağımlılığa örnek olarak

boyu daha uzun olanların ağırlıkları daha fazladır

verilebilir. Burada G_1 "boyu daha uzun olanlar" ve G_2 "ağırlıkları fazladır" ifadelerine karşılık gelmektedir. \mathcal{T} 'nin hesaplanması için ilk olarak veritabanındaki nesne sayısında kurallar oluşturulur. İlgili R_k kurallarının yapısı,

o_k 'nin boyundan boyu uzun olanların ağırlıkları o_k 'nin ağırlığından fazladır veya o_k 'nin boyundan boyu kısa olanların ağırlıkları o_k 'nin ağırlığından azdır

şeklinde dir. R_k kurallarının doğruluk değerleri hesaplanır ve ortalaması alınarak dereceli bağımlılığın doğruluk değeri hesaplanır. R_k kuralının doğruluk değeri T_k aşağıdaki gibi ifade edilir. [1]

$$T_k = \sum_{id} (o_i \in DB | (G_1(o_k, o_j) \wedge G_2(o_k, o_j)) \vee (G_1'(o_k, o_j) \wedge G_2'(o_k, o_j))) \quad (3.10)$$

Burada $G_1(o_k, o_j)$ " $o_k.boy < o_j.boy$ "'u ifade etmektedir. $G_1'(o_k, o_j)$ $G_1(o_k, o_j)$ ifadesinin tersi olup " $o_k.boy > o_j.boy$ "'u ifade etmektedir. Benzer şekilde

$G_2(o_k, o_j) = "o_k.agirlik < o_j.agirlik "$ ve $G_2'(o_k, o_j) = "o_k.agirlik > o_j.agirlik "$ şeklinde dir. Bu eşleştirme lerle (3.10),

$$T_k = \sum_{Id} (o_i \in DB | ((o_k.boy > o_k.boy) \wedge (o_k.agirlik > o_k.agirlik)) \vee \quad (3.11)$$

$$((o_k.boy < o_k.boy) \wedge (o_k.agirlik < o_k.agirlik)))$$

şeklinde elde edilir.

G_i ifadelerinde geçen “<” ve “>” operatörleri doğru/yanlış değerleri veren mantık fonksiyonlardır. Bulanık GFD kuraları elde etmek için ilgili operatörler bulanık operatörler olan “<≈” ve “>≈” ile değiştirilirler. (3.11) ‘deki operatörler bulanık operatörlerle değiştirildiğinde,

$$T_k = \sum_{Id} (o_i \in DB | ((o_k.boy > o_k.boy) \wedge (o_k.agirlik > \approx o_k.agirlik)) \vee \quad (3.12)$$

$$((o_k.boy < o_k.boy) \wedge (o_k.agirlik < \approx o_k.agirlik)))$$

elde edilir. Burada sadece sonuç terimlerini bulanıklaştırıldığına dikkat edilmiştir. Bunun nedeni şart teriminin kendisinin ve tersinin olumlu değer alması durumunda yanlış doğruluk değerlerine ulaşmaya sebebiyet verecek olmasıdır.

Şart yada sonuç kısmında birden fazla terimin “ve” operatörüyle bağlanması durumunda şart önermesini tersi elde edilirken tüm terimlerin tersi alınarak “ve” operatörü ile işlem sokulur. Şart terimi birden fazla terimden oluşuyorsa doğruluk değerinin hesaplanması sırasında sadece şartın kendisi G_i ve tersi G_i' ‘i sağlayan nesnelere üzerinde hesaplama işlemi yapılır, diğer nesnelere ihmal edilir. Bu durumda kural yapısı

$$\text{eğer } G_1(o_k, o_j) \vee G_1'(o_k, o_j)$$

$$\text{ise } (G_1(o_k, o_j) \vee G_2(o_k, o_j)) \text{ veya } (G_1'(o_k, o_j) \vee G_2'(o_k, o_j))$$

şeklinde elde edilir. Kuralın doğruluk değeri de,

$$T_k = \sum_{Id} (o_j \in \{o_i \in DB | G_1(o_k, o_j) \vee G_1'(o_k, o_j)\}) \quad (3.13)$$

$$(G_1(o_k, o_j) \wedge G_2(o_k, o_j)) \vee (G_1'(o_k, o_j) \wedge G_2'(o_k, o_j))$$

olarak elde edilir.

Tablo 3.1 'de verilen örnek veritabanı üzerinde “Alan1 büyük ise Alan2 küçüktür” kuralının doğruluk değerini hesaplayalım. Hesaplama işlemi sırasında kullanılacak bulanık operatörler için aşağıdaki fonksiyonları kullanalım

$$\text{Büyüktür veya benzerdir } (>\approx) : \max((4-|x-y|)/4, 0), x>y)$$

$$\text{Küçüktür veya benzerdir } (<\approx) : \max((4-|x-y|)/4, 0), x<y)$$

Operatörleri incelediğimizde iki elemanın arasındaki fark 4'ten küçük ise küçüktür yada büyüktür şartı sağlanmasa dahi önermenin 0'dan büyük bir doğruluk değeri alması sağlanarak klasik “<” ve “>” operatörleri bulanık hale getirilmiştir. (3.13) 'teki Gönermeleri aşağıdaki gibi şekillenir.

$$G_1(o_k, o_j) : o_k.\text{Alan1} > o_j.\text{Alan1}$$

$$G_1'(o_k, o_j) : o_k.\text{Alan1} < o_j.\text{Alan2}$$

$$G_2(o_k, o_j) : o_k.\text{Alan2} < o_j.\text{Alan2}$$

$$G_2'(o_k, o_j) : o_k.\text{Alan1} > o_j.\text{Alan2}$$

Kuralları ve doğruluk değerlerini aşağıdaki gibi elde ederiz

$$R_k : \text{eğer } ((o_k.\text{Alan1} > o_i.\text{Alan1}) \vee (o_k.\text{Alan1} < o_i.\text{Alan1})) \text{ ise} \\ ((o_k.\text{Alan1} \triangleright o_i.\text{Alan1}) \wedge (o_k.\text{Alan2} <\approx o_i.\text{Alan2})) \vee \\ ((o_k.\text{Alan1} > o_i.\text{Alan1}) \wedge (o_k.\text{Alan1} < o_i.\text{Alan2}))$$

Tablo 3.3: GFD hesaplaması için gerekli ara hesaplamalar

o_k	T_{kj}	
o_1	eğer $((1>1) \vee (1<1))$ ise $((1>1) \wedge (1<\approx 1)) \vee ((1<1) \wedge (1>\approx 1))$	eğer 0 ise 0
	eğer $((1>2) \vee (1<2))$ ise $((1>2) \wedge (1<\approx 4)) \vee ((1<2) \wedge (1>\approx 4))$	eğer 1 ise 0.25
	eğer $((1>4) \vee (1<4))$ ise $((1>4) \wedge (1<\approx 6)) \vee ((1<4) \wedge (1>\approx 6))$	eğer 1 ise 0
	eğer $((1>8) \vee (1<8))$ ise $((1>8) \wedge (1<\approx 6)) \vee ((1<8) \wedge (1>\approx 6))$	eğer 1 ise 0
o_2	eğer $((2>1) \vee (2<1))$ ise $((2>1) \wedge (4<\approx 1)) \vee ((2<1) \wedge (4>\approx 1))$	eğer 1 ise 0.25
	eğer $((2>2) \vee (2<2))$ ise $((2>2) \wedge (4<\approx 4)) \vee ((2<2) \wedge (4>\approx 4))$	eğer 0 ise 0
	eğer $((2>4) \vee (2<4))$ ise $((2>4) \wedge (4<\approx 6)) \vee ((2<4) \wedge (4>\approx 6))$	eğer 1 ise 0.5
	eğer $((2>8) \vee (2<8))$ ise $((2>8) \wedge (4<\approx 6)) \vee ((2<8) \wedge (4>\approx 6))$	eğer 1 ise 0.5
o_3	eğer $((4>1) \vee (4<1))$ ise $((4>1) \wedge (6<\approx 1)) \vee ((4<1) \wedge (6>\approx 1))$	eğer 1 ise 0

	eğer ((4>2) \vee (4<2)) ise ((4>2) \wedge (6< \approx 4)) \vee ((4<2) \wedge (6 \approx 4))	eğer 1 ise 0.5
	eğer ((4>4) \vee (4<4)) ise ((4>4) \wedge (6< \approx 6)) \vee ((4<4) \wedge (6 \approx 6))	eğer 0 ise 0
	eğer ((4>8) \vee (4<8)) ise ((4>8) \wedge (6< \approx 6)) \vee ((4<8) \wedge (6 \approx 6))	eğer 1 ise 1
o_4	eğer ((8>1) \vee (8<1)) ise ((8>1) \wedge (6< \approx 1)) \vee ((8<1) \wedge (6 \approx 1))	eğer 1 ise 0
	eğer ((8>2) \vee (8<2)) ise ((8>2) \wedge (6< \approx 4)) \vee ((8<2) \wedge (6 \approx 4))	eğer 1 ise 0.5
	eğer ((8>4) \vee (8<4)) ise ((8>4) \wedge (6< \approx 6)) \vee ((8<4) \wedge (6 \approx 6))	eğer 1 ise 1
	eğer ((8>8) \vee (8<8)) ise ((8>8) \wedge (6< \approx 6)) \vee ((8<8) \wedge (6 \approx 6))	eğer 0 ise 0

T_i değerleri aşağıdaki gibi elde edilir.

$$T_1 = \frac{0.25 + 0 + 0}{1 + 1 + 1} = 0.08$$

$$T_2 = \frac{0.25 + 0.5 + 0.5}{1 + 1 + 1} = 0.42$$

$$T_3 = \frac{0 + 0.5 + 1}{1 + 1 + 1} = 0.5$$

$$T_4 = \frac{0 + 0.5 + 1}{1 + 1 + 1} = 0.5$$

Son olarak “A an1 büyük ise A an2 küçüktür” kuralı için \mathcal{T}

$$\mathcal{T} = \frac{T_1 + T_2 + T_3 + T_4}{4} = \frac{0.08 + 0.42 + 0.5 + 0.5}{4} = 0.375$$

elde edilir.

4. GELİŞTİRİLEN UYGULAMA

Yüksek lisans çalışmasında, ilişkisel veritabanlarındaki fonksiyonel ve dereceli bulanık ilişkilerin doğruluk değerlerini sorgulayan ve sonuçları kaydeden bir uygulamanın geliştirilmesi hedeflenmiştir. Bu amaçla yönelik olarak, istemci-sunucu(client-server) mimarisinde bir uygulama geliştirilmiştir. Geliştirilen uygulama öncelikle ORACLE veritabanı sistemleri üzerinde sorgulama yapacak şekilde tasarlanmakla birlikte, ODBC(open database connectivity) bağlantısı ile diğer beynelânel sistemler üzerinde de bulanık sorgulama yapabilir.

Yazılımın analizinde ve geliştirilmesinde, sistemin, bulanık küme teorisi ve bulanık sorgulamaya hakim olan kişiler tarafından da rahatlıkla kullanılabilir nitelikte olmasına dikkat edilmiştir. Bu amaçla yönelik olarak, ekranların tasarımında anlaşılabilirliğe önem verilmiştir.

Yazılımın geliştirilmesi sırasında üzerinden geçilen aşamalar bu bölümde açıklanacaktır.

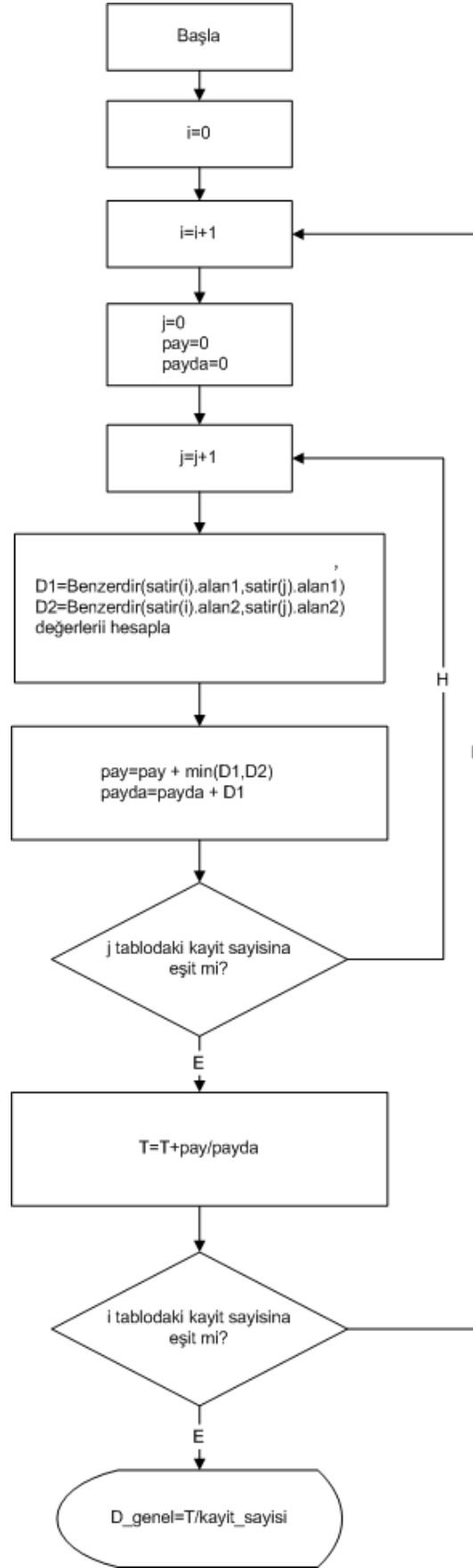
4.1 Kullanılan Algoritmalar ve Akış Diyagramları

Dereceli ve fonksiyonel bulanık bağımlılıkların doğruluk derecelerinin tespiti sürecinde Bölüm 3 'de tarif edilen ve (3.7),(3.9) 'da formüle edilen sistematik yapı kullanılmıştır.

4.1.1 Bulanık Fonksiyonel İlişki Hesabı için Algoritma

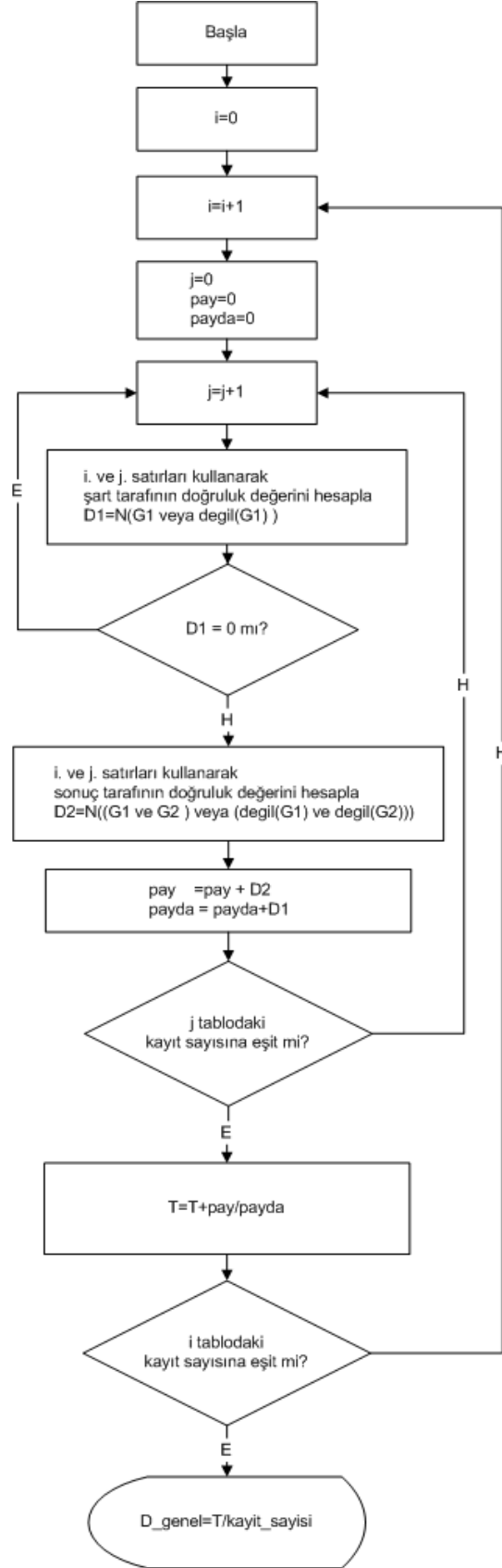
Bulanık fonksiyonel ilişki doğruluk değeri hesabı akış diyagramı Şekil 4.2 'de gösterilmiştir. Bu akışa göre algoritma içi içe dönen iki döngüden oluşmaktadır. Döngüler analizi yapılan tablodaki kayıt sayısı kadar tekrardan oluşmaktadır.

Farkedileceği gibi (3.7) 'de adı geçen $T()$ fonksiyonu olarak minimum fonksiyonu kullanılmıştır. Akış diyagramında geçen $benzerdir()$ ve $degil()$ fonksiyonları benzerlik ve degil operatörlerine karşılık gelen doğruluk değerlerini döndüren fonksiyonlardır.



Şekil 4 1 : Bulanık Fonksiyonel İlişkiler için Doğruluk Değeri Hesabı Akış Diyagramı

4.1.2 Dereceli Bulanık İlişki Hesabı için Algoritma



Şekil 4.2: Dereceli Bulanık İlişkiler için Doğruluk Değeri Hesabı Akış Diyagramı

Dereceli bulanık ilişki doğruluk değeri hesabı akış diagra m Şekil 4.2 ‘de gösterilmiştir. Bu akışa göre algoritma içi içe dönen iki döngüden oluşmaktadır. Döngüler analizi yapılan tablodaki kayıt sayısı kadar tekrardan oluşmaktadır

Akış di gramında geçen benzerdir() ve degil() fonksiyonları benzerlik ve degil operatörlerine karşılık gelen doğruluk değerlerini döndüren fonksiyonlardır. Akış di agar mında geçen G2 değerlerini hesaplanması sırasında kullanılacak “büyük tür veya benzerdir” , “küçük tür veya benzerdir” fonksiyonlarının yazılımda hangi şekilde kullanılacağı 4.2 ‘de açıklanacaktır.

4.2 Yazılı mda Uygulanan Bulanık Fonksiyonlar

Dereceli ve fonksiyonel bulanık ilişkilerin sorgulanması sürecinde gerekli olan “benzerdir”, “büyük tür veya benzerdir”, “küçük tür veya benzerdir” bulanık fonksiyonlarının seçimi, geliştirilen uygulamanın içerisi de belirli sınırlar dahilinde esnek bırakılmıştır. Aşağıda listelenen fonksiyonların herbirinin eşik değerleri belirlenebildiğinden, amaca yönelik olarak sonsuz sayıda bulanık fonksiyon kullanılabilir.

Kullanılabilen bulanık fonksiyonlar aşağıdaki gibidir. Fonksiyonlar, geliştirilen uygulama içerisinde, fonksiyon tipi (tirgonal, dikey parabolik, yatay parabolik) ve fark tipi (fark, oran) şeklinde iki parametrenin kombinasyonları şeklinde birbirlerinden ayrılırlar.

Benzerdir

$$\mu(x \approx y) = \begin{cases} 0 & |x - y| > eşik \\ \frac{eşik - |y - x|}{eşik} & 0 \leq |x - y| \leq eşik \end{cases} \quad (4.1)$$

$$\mu(x \approx y) = \begin{cases} 0 & \lambda < eşik \\ \frac{(1 - \lambda) - (1 - eşik)}{(1 - eşik)} & eşik \leq \lambda \leq 1 \end{cases} \quad (4.2)$$

$$\lambda = \text{Min}(x, y) / \text{Max}(x, y)$$

$$\mu(x \approx y) = \left\{ \begin{array}{ll} 0 & |x - y| > eşik \\ \frac{(eşik - |x - y|)^2}{eşik^2} & 0 \leq |x - y| \leq eşik \end{array} \right\} \quad (4.3)$$

$$\mu(x \approx y) = \left\{ \begin{array}{ll} 0 & \lambda < eşik \\ \frac{((1 - eşik) - (1 - \lambda))^2}{(1 - eşik)^2} & eşik \leq \lambda \leq 1 \end{array} \right\} \quad (4.4)$$

$\lambda = \text{Min}(x, y) / \text{Max}(x, y)$

$$\mu(x \approx y) = \left\{ \begin{array}{ll} 0 & |x - y| > eşik \\ \frac{\sqrt{eşik - |x - y|}}{\sqrt{eşik}} & 0 \leq |x - y| \leq eşik \end{array} \right\} \quad (4.5)$$

$$\mu(x \approx y) = \left\{ \begin{array}{ll} 0 & \lambda < eşik \\ \frac{\sqrt{(1 - eşik) - (1 - \lambda)}}{\sqrt{1 - eşik}} & eşik \leq \lambda \leq 1 \end{array} \right\} \quad (4.6)$$

$\lambda = \text{Min}(x, y) / \text{Max}(x, y)$

Büyük tür veya Benzerdir

$$\mu(x > \approx y) = \left\{ \begin{array}{ll} 0 & y - x > eşik \\ \frac{eşik - |y - x|}{eşik} & 0 \leq y - x \leq eşik \\ 1 & x > y \end{array} \right\} \quad (4.7)$$

$$\mu(x > \approx y) = \left\{ \begin{array}{ll} 0 & x / y < eşik \\ \frac{(1 - x / y) - (1 - eşik)}{(1 - eşik)} & eşik \leq x / y \leq 1 \\ 1 & x \geq y \end{array} \right\} \quad (4.8)$$

$$\mu(x > \approx y) = \left\{ \begin{array}{ll} 0 & y - x > eşik \\ \frac{(eşik - |y - x|)^2}{eşik^2} & 0 \leq y - x \leq eşik \\ 1 & x > y \end{array} \right\} \quad (4.9)$$

$$\mu(x > \approx y) = \left\{ \begin{array}{ll} 0 & x / y < eşik \\ \frac{((1 - eşik) - (1 - x / y))^2}{(1 - eşik)^2} & eşik \leq x / y \leq 1 \\ 1 & x \geq y \end{array} \right\} \quad (4.10)$$

$$\mu(x > \approx y) = \left\{ \begin{array}{ll} 0 & y - x > eşik \\ \frac{\sqrt{eşik - |y - x|}}{\sqrt{eşik}} & 0 \leq y - x \leq eşik \\ 1 & x > y \end{array} \right\} \quad (4.11)$$

$$\mu(x > \approx y) = \left\{ \begin{array}{ll} 0 & x / y < eşik \\ \frac{\sqrt{(1 - eşik) - (1 - x / y)}}{\sqrt{(1 - eşik)}} & eşik \leq x / y \leq 1 \\ 1 & x > y \end{array} \right\} \quad (4.12)$$

Küçüktür veya Benzerdir

$$\mu(x < \approx y) = \left\{ \begin{array}{ll} 0 & x - y > eşik \\ \frac{eşik - |x - y|}{eşik} & 0 \leq x - y \leq eşik \\ 1 & x < y \end{array} \right\} \quad (4.13)$$

$$\mu(x \approx y) = \begin{cases} 0 & y/x < eşik \\ \frac{(1 - y/x) - (1 - eşik)}{(1 - eşik)} & eşik \leq y/x \leq 1 \\ 1 & x < y \end{cases} \quad (4.14)$$

$$\mu(x \approx y) = \begin{cases} 0 & x - y > eşik \\ \frac{(eşik - |x - y|)^2}{eşik^2} & 0 \leq x - y \leq eşik \\ 1 & x < y \end{cases} \quad (4.15)$$

$$\mu(x \approx y) = \begin{cases} 0 & y/x < eşik \\ \frac{((1 - eşik) - (1 - y/x))^2}{(1 - eşik)^2} & eşik \leq y/x \leq 1 \\ 1 & x < y \end{cases} \quad (4.16)$$

$$\mu(x \approx y) = \begin{cases} 0 & x - y > eşik \\ \frac{\sqrt{eşik - |x - y|}}{\sqrt{eşik}} & 0 \leq x - y \leq eşik \\ 1 & x < y \end{cases} \quad (4.17)$$

$$\mu(x \approx y) = \begin{cases} 0 & y/x < eşik \\ \frac{\sqrt{(1 - eşik) - (1 - y/x)}}{\sqrt{1 - eşik}} & eşik \leq y/x \leq 1 \\ 1 & x < y \end{cases} \quad (4.18)$$

4.3 Geliştirme Ortam ve Kullanılan Temel Yazılım Yapıları

Yazılım Windows 2000 işletim sistemi üzerinde Oracle Developer 6i uygulamaya geliştirme aracı ile geliştirilmiştir. Temel veritabanı, Oracle 8.1.7.0 Personal Edition olarak kabul edilerek yazılım geliştirilmiştir. Ancak uygulamaya Oracle 7 ve üzeri tüm veritabanı sürümleri ile uyumlu çalışabilecek yapıdadır. Oracle dışı veritabanlarına

erişimin yapılabilmesi için uygulamanın tekrar düzenlenmesi yada ilgili Oracle dışı veritabanı tablolarının bir Oracle veritabanı içerisinden görünür duruma getirilmesi gerekmektedir. Geliştirilen “Fonksiyonel ve Dereceli Bulanlık İlişki Bulma Aracı” Oracle veritabanları ile Sql-Net ile iletişime kurmaktadır.

4.3.1 Kullanılan Temel Veritabanı Birimleri

Veritabanında kullanıcının hakkı olan tabloların listelenmesi ve kullanıcının analiz etmek istediği tablonun alanlarının listesi için Oracle veritabanlarında standart olarak bulunan “user_tables” ve “user_tab_columns” view’ları kullanılmaktadır.

User_tables: Bu view veritabanına bağlı olunan kullanıcının şemasında(schema) bulunan tabloların listesini ve temel özelliklerine ulaşmada kullanılır. Burada dikkat edilecek husus, kullanıcının şemasında sadece kendi yarattığı tabloların olduğudur. Dolayısıyla bu view ile sadece sisteme giriş yapan kullanıcıya ait tablolara ulaşılabilir. Bu özellik sayesinde diğer kullanıcıların tablolarının analizi önlenmiş olur. Bu view’ın geliştirilen uygulamada kritik öneme sahip alanı “table_name” alanıdır.

User_tab_columns: Bu view veritabanına bağlı olunan kullanıcının şemasına ait tablolardaki alanlarının listesi ve temel özelliklerine erişimde kullanılır. “user_tab_columns” view’ı ile kullanıcı, sahip olduğu belirli bir tablonun hangi alanlardan oluştuğu ve alanların yapısı bilgisini edinebilir. Bu view’ın geliştirilen uygulamada kritik öneme sahip alanları “table_name”, “column_name” ve “data_type” alanlarıdır.

4.3.2 Kullanılan Temel PL/SQL Yapıları

Oracle Developer 6i içerisinde kullanılan kodlama dili bir çok Oracle ürününde olduğu gibi PL/SQL ‘dir. PL/SQL veritabanı işleme sürecinde etkili sorgulama işlemleri ile prosedürel bir yazılımdilinin birçok özelliklerini bir arada barındıran bir kodlama dili olarak karşımıza çıkmaktadır. Uygulamada etkin olarak kullanılan temel PL/SQL yapıları aşağıda tarif edilmiştir.

Paketler (Packages): Paketler birbirleriyle mantıksal yada işlevsel olarak ilişkili nesne, prosedür ve fonksiyonları gruplayan yapılardır. Paketler iki temel parçadan oluşurlar; gövde(body) ve tanım(specification). Oracle veritabanları ve Oracle Forms ile yapılan çalışmalarda paketler veritabanında, yada forms içerisinde yaratılabilirler.

Veritabanı üzerinde yaratılacak paketin tüm uygulamalar tarafından ulaşılabilir özelliğe olduğu unutulmamalıdır.

Tam bölümü paketi arayüzü olarak görev yapar; bu bölümde paket içerisinde yer alacak değişken, sabit, fonksiyon, prosedür ve diğer nesnelerin tanımlanması için içerir. Gövde bölümü, tam bölümünde tanımlanan nesnelere içerik kodlarının bulunduğu yapıdır. Tam bölümünde tanımlanan tüm yapıların gövde bölümünde yer alması gerekir, ancak, tam bölümünde tanımlanan herhangi bir nesnenin içerik kodu gövde kısmında yer almaz.

Paketler uygulamalar arası veri transferi için de kullanılabilirler. Veritabanı üzerinde tanımlanan paketlerdeki paket değişkenleri değerlerini oturum boyunca korurlar. Dolayısıyla, oturum boyunca saklanması öngörülen değerler paket değişkenlerinde saklanabilirler.

Dbms_Sql Paketi: Standart PL/SQL cümleleri sabit yapıya sahiptirler. Yani oluşturulan cümleler geliştirilebilir şekilde hazırlanır, ve bu cümleler çalıştırma aşamasında aynen çalıştırılırlar. Bu durum tasarım sırasında bilinmeyen bir tablodan yapılacak olan seçme yada önceden bilinmeyen bir prosedürün çalıştırılması gibi dinamik davranışların icra edilmesini engeller. Oracle veritabanlarının üzerinde standart olarak oluşturulmuş olan Dbms_Sql paketi bu gibi sınırların giderilmesi amacıyla kullanılır.

Dbms_Sql paketinin geliştirilenez uygulaması içerisinde uygulanan genel kullanım adımları ile paketteki fonksiyon ve prosedürleri aşağıdaki gibidir.

CURSOR AÇMA (OPEN_CURSOR): Dbms_Sql paketi ile bir SQL cümlesi çalıştırmanın ilk şartı açık bir cursorün bulunmasıdır. “OPEN_CURSOR” fonksiyonu çağrıldığında ilgili fonksiyon açılan cursor numarasını döndürür. Bu cursor sadece Dbms_Sql paketi tarafından kullanılabilir.

ANLAMLANDIRMA (PARSE): SQL cümlesi çalıştırılmadan önce “PARSE” prosedürü ile anlamlandırılmasıdır. Bu işlem sırasında veritabanı cümlesinin yazımını kontrol eder ve cümle ile cursor arasında ilişkisinin kurulmasını sağlar.

ALANLARIN TANIMLANMASI (DEFINE_COLUMN) : “Select” cümlesi içerisinde seçilecek alanlar “DEFINE_COLUMN” prosedürü ile tanımlanır. Cümlesinin seçim listesindeki alanlar seçime sıralarına göre birbirlerinden ayırt edilirler.

CURSOR' UN ÇALIŞTIRILMASI: (EXECUTE) : Yaratılan, anlamlandırılan ve alanları tanımlanan cursor “EXECUTE” fonksiyonu ile çalıştırılır.

SATIRLARIN ÇEKİLMESİ (EXECUTE_ROWS): Yapılan sorgu karşılığında dönen satırlar “FETCH_ROWS” fonksiyonu ile çekilirler. İlgili fonksiyonun her çağrışı bir adet yeni satırı veri tabanından getirir.

ALAN DEĞERLERİNİN OKUNMASI (COLUMN_VALUE) : Çekilen satırların değerleri “COLUMN_VALUE” prosedürü ile okunurlar.

CURSOR' UN KAPATILMASI (CLOSE_CURSOR) : Ortam içerisinde kullanım sürecini dolduran cursor'ler “CLOSE_CURSOR” prosedürü ile kapatılırlar. Bu işlem ile cursor için rezerve edilen tüm sistem kaynakları serbest bırakılmış olur.

Geliştirilen tez projesinin fonksiyonel ve dereceli bulanlık ilişki bulma prosedürlerinde Dbms_Sql paketi kullanılmıştır. Buradaki kullanıma amacı, tasarımı aşamasında belirsiz olan tablolara ve bu tabloların kolonlarına ulaşımın sağlanabilmesidir.

4.4 Geliştirilen Birimler

Çalışma sırasında aşağıda açıklanan form ve menu birimleri geliştirilmiştir.

f_analyzer_main.fb : Bu sistemin ana giriş sayfasıdır. Üzerinde yazılımhakkında temel tanıtıcı bilgiler olan bir canvas ile sistemin ana menüsü yer alır.

f_analyzer_main.mmb : Bu sistemin ana menüsüdür. Ana sayfanın üzerinde görüntülenir.

f_fuzzy_analyzer.fb : Bu birim bulanlık fonksiyonel analiz işlemlerini yürüten formdur.

g_fuzzy_analyzer.fb : Dereceli fonksiyonel analiz işlemlerini yürüten formdur.

f_logdan_dosyaya.fb : Kaydedilen geçmiş sorgu sonuçlarının belirlenen tarih aralığı kısıtlaması ile dosya sistemi üzerine çıkarılmasını sağlayan formdur.

f_log_temizle.fb : Kaydedilen geçmiş sorgu sonuçlarının belirlenen tarih aralığı kısıtlaması ile temizlenmesini sağlayan formdur.

4.4.1 Uygulamanın Oluşturduğu Tablolar

Yapılan fonksiyonel ve dereceli sorguların sonuçları veritabanı üzerindeki tablolarda kayıt altına alınmaktadır. Uygulamanın ana menüsündeki “Geçmiş Sorgular” altındaki uygulamalar ile bu kayıtlar, istenir ise dosya sistemi üzerinde istenen yere dosya olarak alınabilmekte ve tevizlenebilmektedir. Bulanık fonksiyonel ve dereceli bulanık sorgu sonuçları ayrı ayrı kaydedilmektedir. Şekil 4.3 ve Şekil 4.4’te geçmiş tutanak tablolarının yapıları gösterilmiştir.

FA_F_LOG	
TABLO_ADI	VARCHAR2(50)
ANALIZ_TARIHI	DATE
KAYIT_SAYISI	NUMBER(8)
ALAN1	VARCHAR2(50)
FONKSIYON_TIPI1	VARCHAR2(30)
FARK_TIPI1	VARCHAR2(1)
FARK_ESIGI1	NUMBER(28,5)
ALAN2	VARCHAR2(50)
FONKSIYON_TIPI2	VARCHAR2(30)
FARK_TIPI2	VARCHAR2(1)
FARK_ESIGI2	NUMBER(28,5)
DOGRULUK_DEGERI	NUMBER(6,5)

Şekil 4.3: Bulanık Fonksiyonel İlişki Sorgu Sonuçlarının Kaydedildiği Tablo

FA_G_LOG_MST		FA_G_LOG_DET	
LOG_ID	NUMBER(5)	LOG_ID	NUMBER(5)
TABLO_ADI	VARCHAR2(50)	SART_SONUC	VARCHAR2(5)
ANALIZ_TARIHI	DATE	ALAN	VARCHAR2(50)
WHERE_CLAUSE	VARCHAR2(2000)	FONKSIYON_TIPI	VARCHAR2(30)
KAYIT_SAYISI	NUMBER(8)	FARK_TIPI	VARCHAR2(1)
SART_OPERATOR	VARCHAR2(3)	KUCUK_BUYUK	VARCHAR2(1)
SONUC_OPERATOR	VARCHAR2(3)		
DOGRULUK_DEGERI	NUMBER(6,5)		

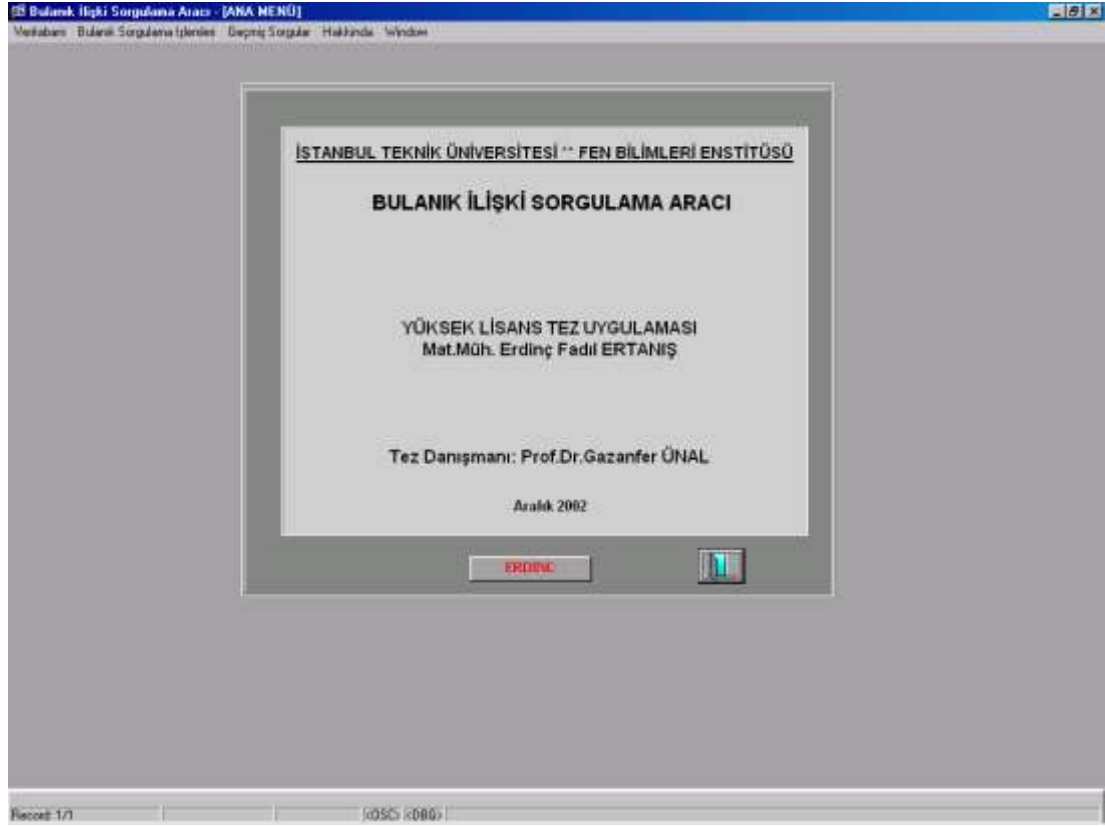
Şekil 4.4: Dereceli Bulanık İlişki Sorgu Sonuçlarının Kaydedildiği Tablo

4.5 Dereceli ve Fonksiyonel Bulanık İlişki Sorgulama Aracının Kullanımı

Geliştirilen uygulamanın kullanımları ve açıklayıcı ekran çıktıları bu bölümde açıklanacaktır.

4.5.1 Genel

Uygulama çalıştırıldığında Şekil 4.5 'te gösterilen ana giriş ekranı ve ana menü görüntülenir.



Şekil 4.5: Uygulama Giriş Ekranı

Menü,

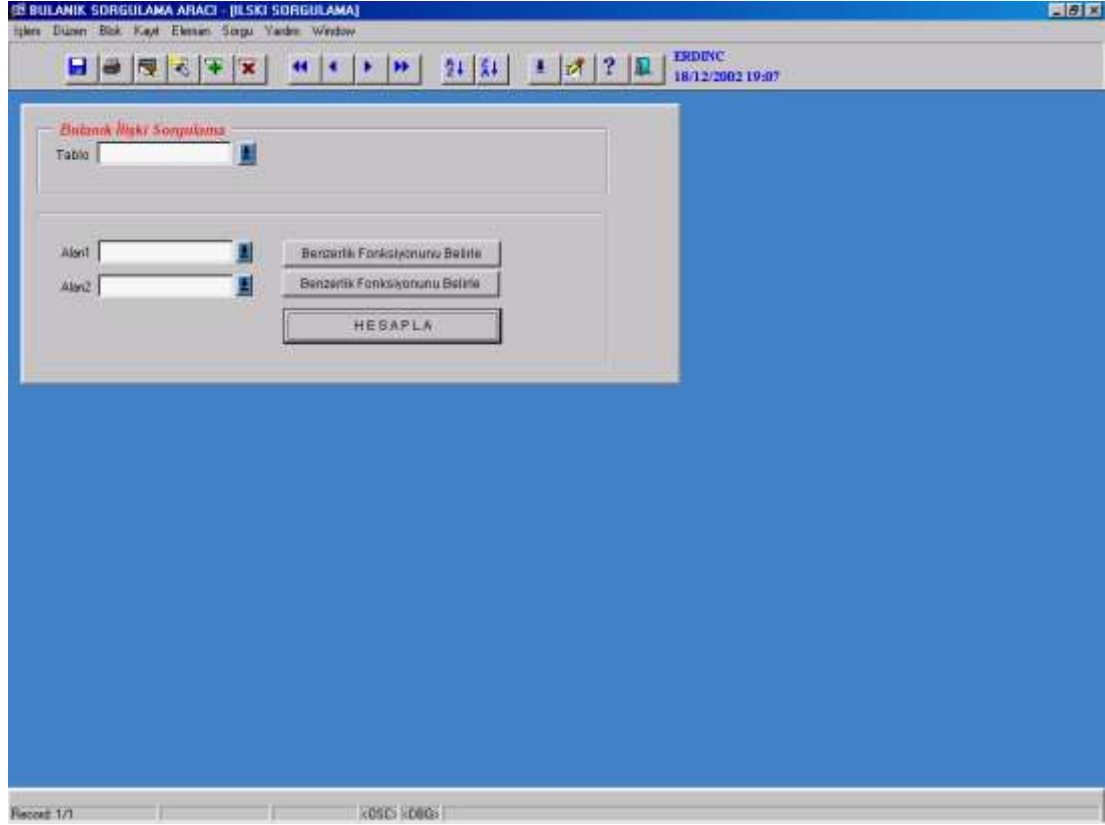
- Veri Tabanı
 - Bağlan
 - Bağlantıyı Kes
- Bulanık Sorgulama İşlemleri
 - Fonksiyonel Bulanık İlişkiler
 - Dereceli Bulanık İlişkiler
- Geçmiş Sorgular
 - Geçmiş Tezlerle
 - Dosyaya Çıkart
- Hakkında
 - Hakkında

hi yerarşisinde oluşturulmuştur.

Herhangi bir ara ekrandan çıktığında Şekil 4.5'te gösterilen ana ekrana dönüş yapılır.

4.5.2 Fonksiyonel Bulanık Sorgulama

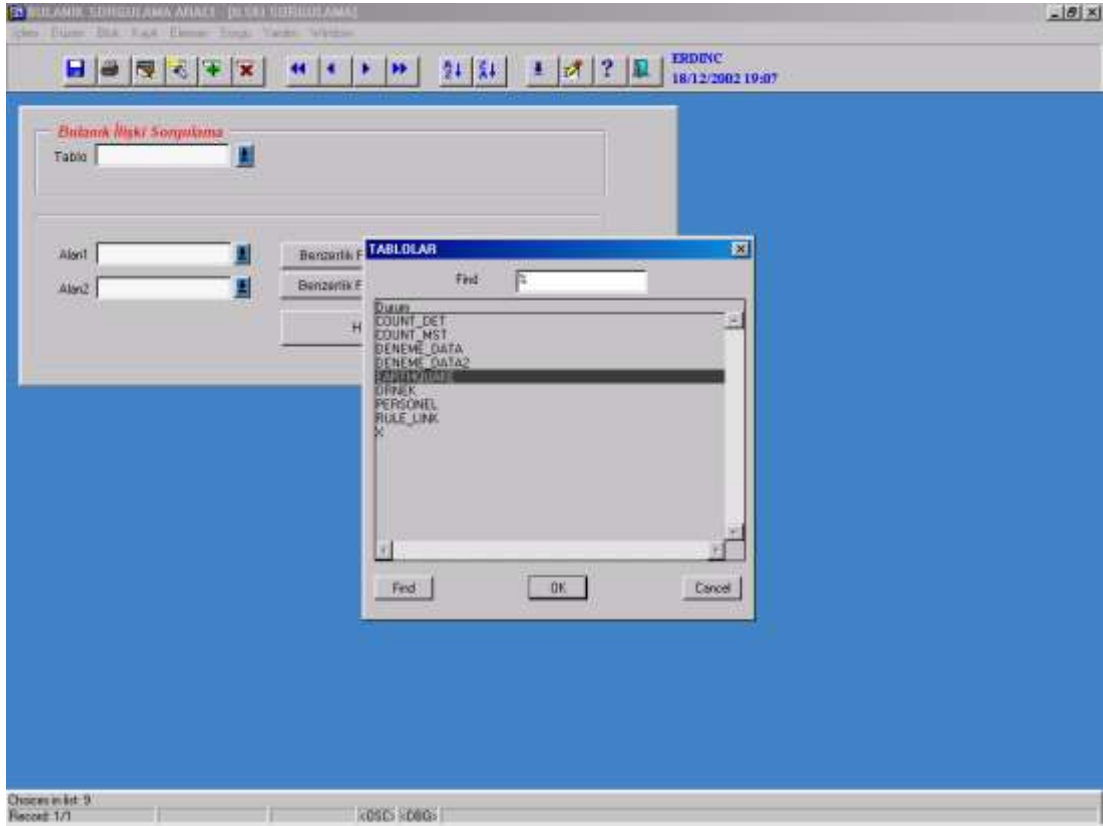
Bulanık Sorgulama İşlemleri ana menüsünden Fonksiyonel Bulanık İlişkiler seçilir. Şekil 4.6 görülmekte olan ekran açılır.



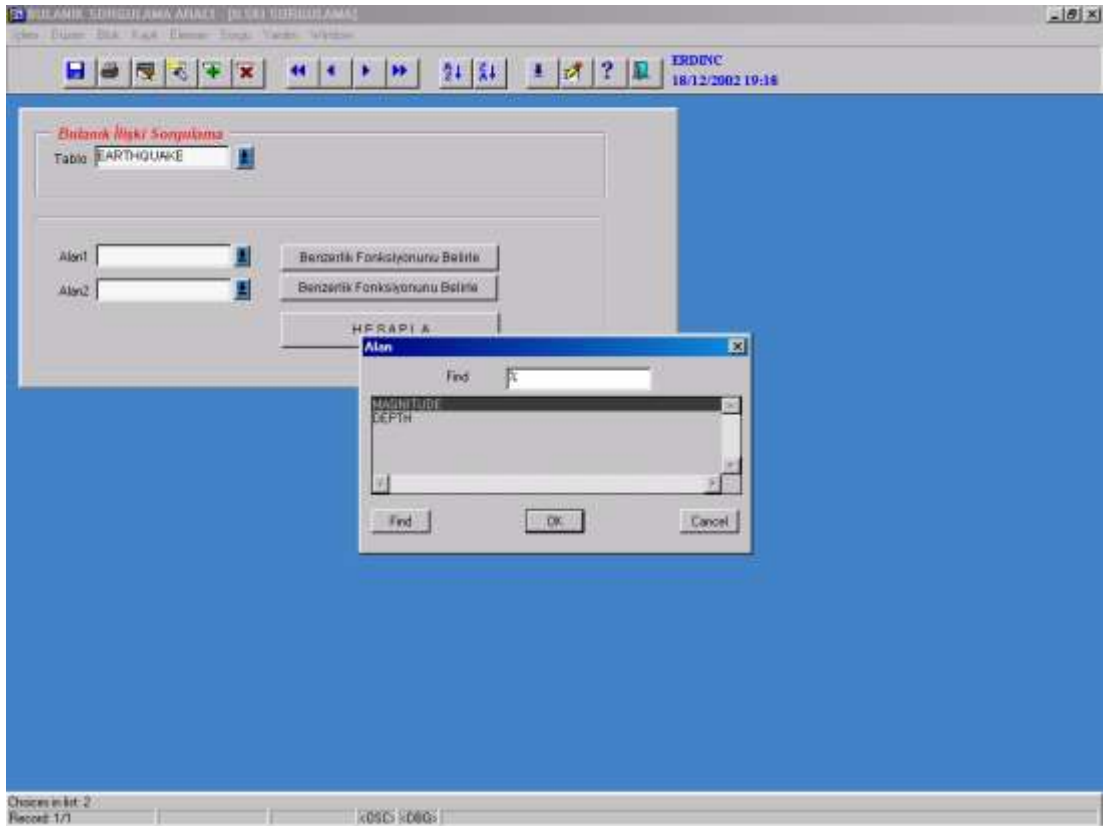
Şekil 4.6 : Fonksiyonel bulanık ilişki sorgulama ekranı

Bu ekranda “Tablo”, “Alan1” ve “Alan2” alanları için değerler seçilir. Bu işlem ilgili alanların yanlarında bulunan düğmelerin açtığı değer listeleri (LOV) ile gerçekleştirilir. “Tablo” alanının değer listesine sadece kendisinin sahip olduğu tablolar, “Alan1” ve “Alan2” alanlarının değer listelerine seçilen tablonun alanları getirilir. “Tablo” alanını seçilmeden “Alan1” yada “Alan2” için seçim yapılmamasına izin verilmeyecektir. Şekil 4.7 ve Şekil 4.8’de değer listelerinin kullanımını gösterilmiştir.

Benzerlik fonksiyonu her bir alan için ayrı ayrı belirlenebilir. Bu işlem için “Benzerlik Fonksiyonunu Belirle” düğmeleri ile açılan seçim pencereleri kullanılır. Şekil 4.9’de görülen pencerede fonksiyon tipi, fark tipi, fark eşiği alanları ile kullanılacak benzerlik fonksiyonunun yapısı belirlenir. Seçim sırasında pencerede fonksiyonun denklemi ve temsili grafiği de gösterilir. Seçim sırasında seçilebilen fonksiyonlar Bölüm 4.2’de ifade edilmiştir.

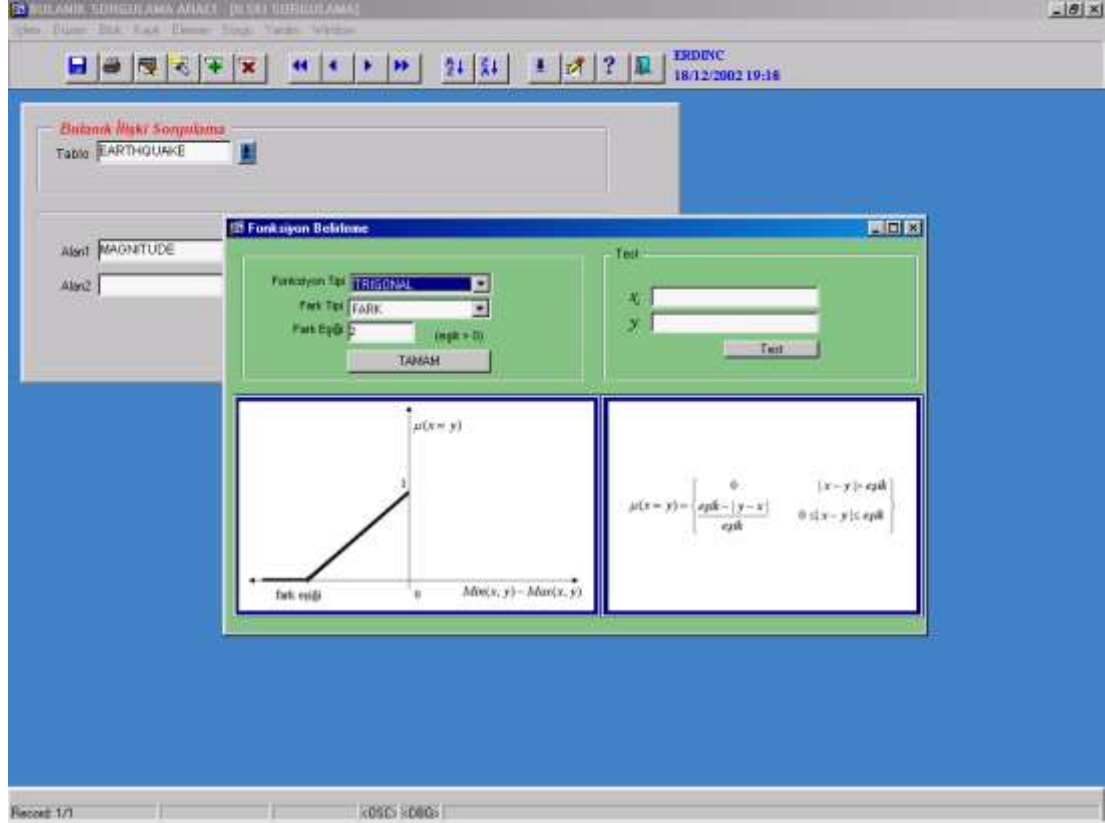


Şekil 4 7:”Tablo” alanı için değer listesi



Şekil 4 8:”Alan1” alanı için değer listesi

Benzerlik fonksiyonu belirleme penceresinin test işlevi ile seçilen benzerlik fonksiyonu gözden geçirilebilir. “Test” düğmesine basıldığında, girilen “x” ve “y” değerleri için, “x benzerdir y” ifadesinin doğruluk değeri görülebilir.



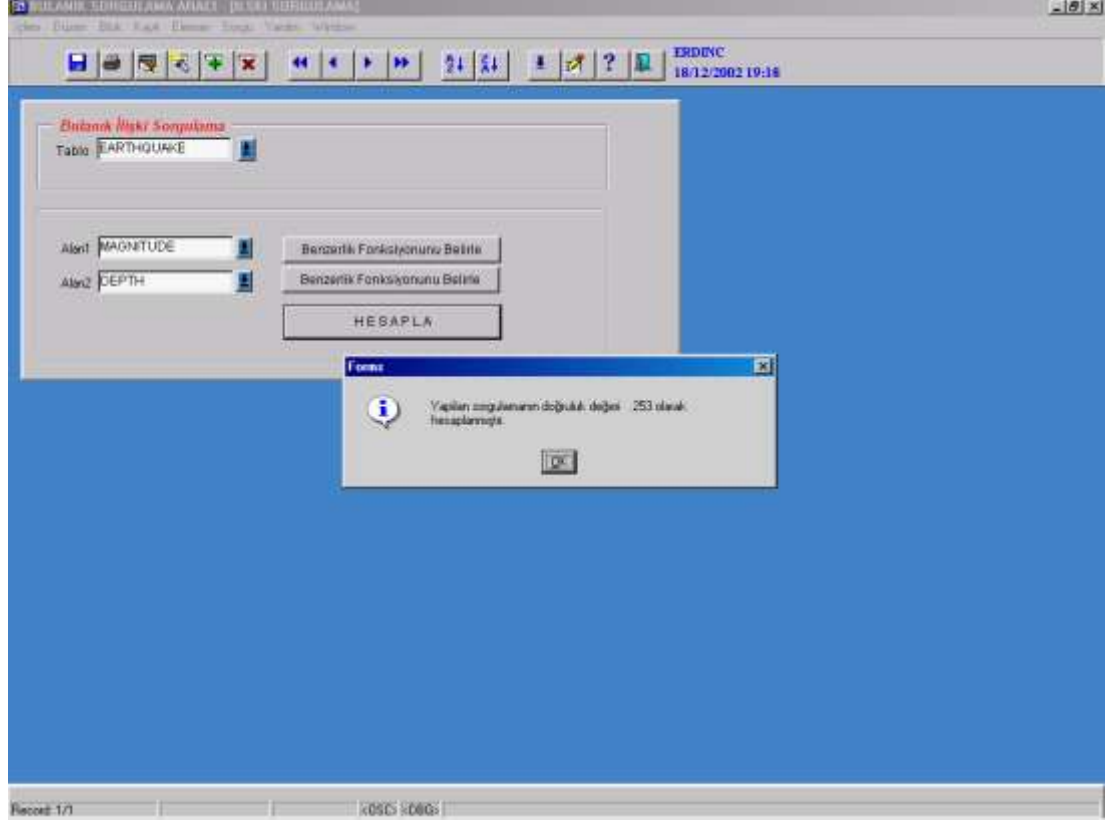
Şekil 4.9: Benzerlik fonksiyonunun seçilmesi

Fark tipi alanı, fonksiyonun yapısını belirleyen en önemli parametre olarak görülebilir. İki sayının benzerliğine, bu sayıların oranına göre karar verildiği durumlarda fark tipi “oran”, farkına göre karar verildiği durumlarda fark tipi “fark” olarak seçilmelidir.

Örneğin sıcaklık bilgisi içeren bir alan için fark tipinin “oran” olarak seçilmesi daha uygun olacaktır. Zira, 15°C ile 17°C nasıl benzer olarak kabul edilebilirse 1500°C ile 1700°C ‘da benzer olarak görülebilir. Burada insan düşüncesinde oluşan ayırt edici kriter, açık bir şekilde iki sayının birbirine oranıdır.

Fonksiyonel bulamak ilişki belirleme ekranındaki tüm alanlar seçilip “HESAPLA” düğmesine basıldığında kullanıcı girilen alan arasındaki fonksiyonel ilişkinin doğruluk değeri konusunda bilgilendirilir. (Şekil 4.10)

Girilen sorgunun yapısı ve sonucu uygulama tarafından veritabanında oluşturulan geçmiş tablolarında saklanır.



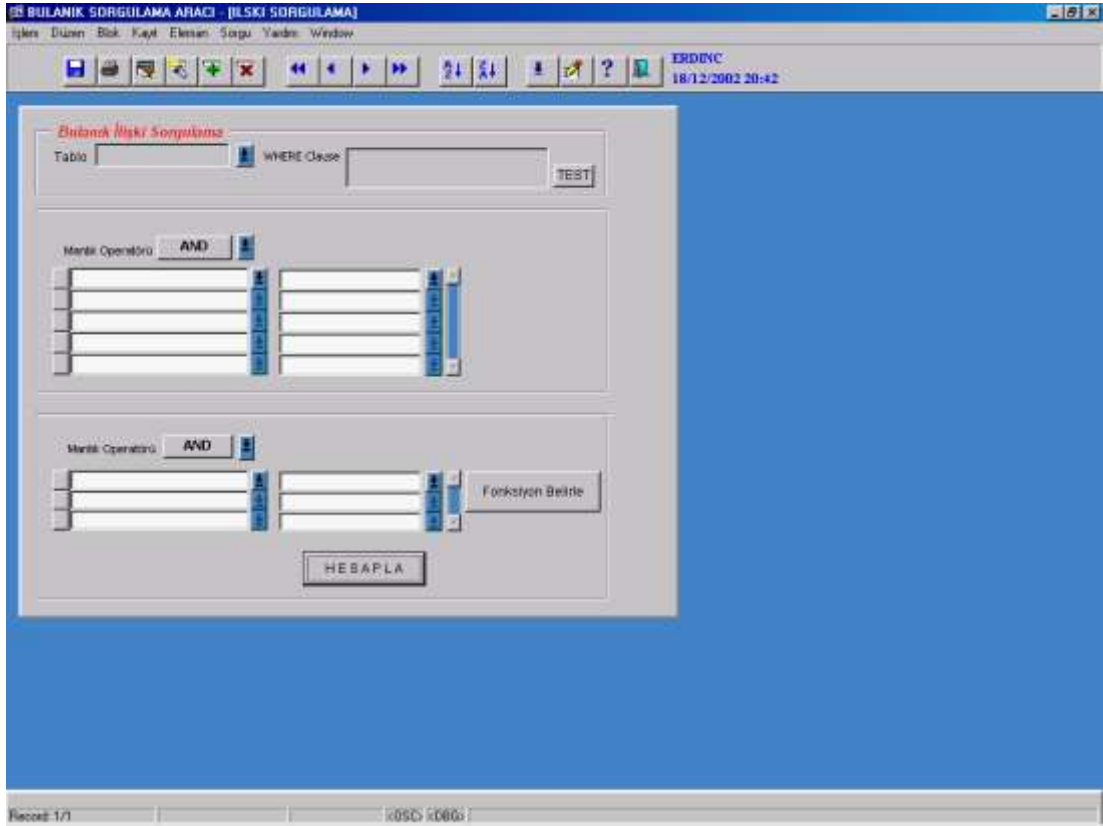
Şekil 4.10: Doğruluk değerinin görüntülenmesi

4.5.3 Dereceli Bulan k İlişki Sorgulama

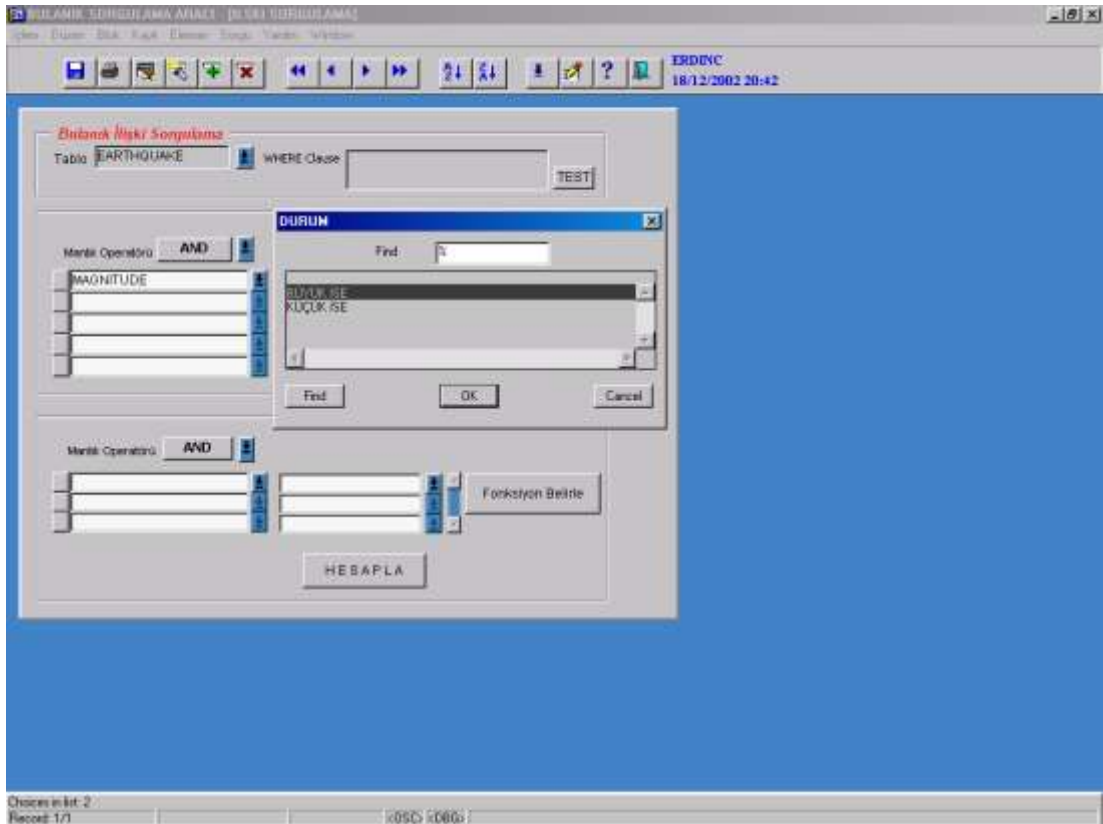
Bulan k Sorgulama İşlemleri ana menüsünden Dereceli Bulan k İlişkiler seçilir. Şekil 4.10’ da görüntülenen ekran açılır. Ekranda bulunan tablo alanı ile şart ve sonuç alanları fonksiyonel ilişki sorgulama ekranındaki sürece benzer şekilde doldurulur.

Bununla birlikte her bir alan için bir operatörün seçilmesi gerekmektedir. Seçim işlemi alanın yanında bulunan ikinci düğme ile yapılacaktır. Şekil 4.12’ de operatör seçimi için kullanılan değer listesinin görüntülenmesi yer almaktadır.

Yapılacak olan kural sorgusunda dilediği kadar şart ve sonuç alanı kullanılabilir. Bir alanın şart ve sonuç bölümlerinde kullanılması mümkün olsada bu şekilde kullanılması tavsiye edilmez.



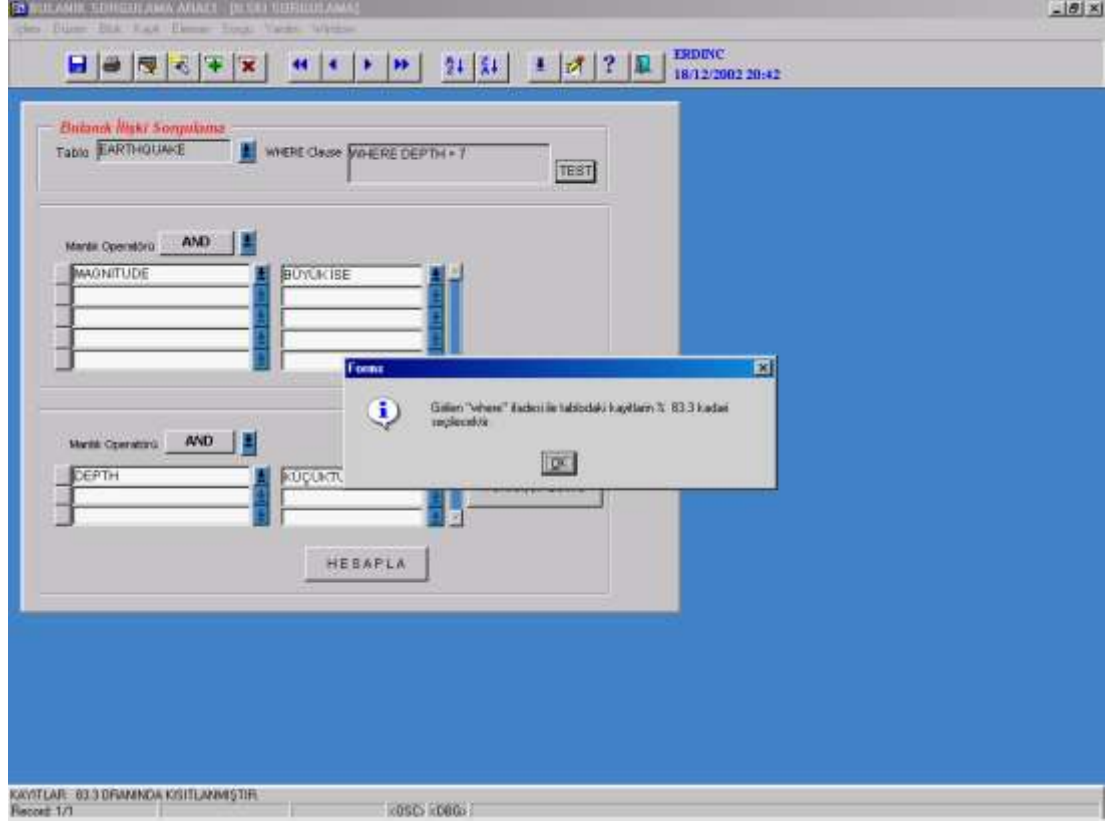
Şekil 4 11: Dereceli bulanık ilişki sorgulama ekranı



Şekil 4 12: Şart alanı için durum operatörü seçilmesi

Şart ve sonuç bloklarının üzerinde yer alan mantıksal operatör ile ilgili bloktaki satırların hangi mantık operatörü ile bağlanacağı belirlenir. Seçmişlemi mantık operatörünün yanındaki düğme ile yapılır.

Ekrandaki “where clause” alanına standart SQL where ifadesi yazılarak seçilen tablo üzerinde kısıtlama yapılabilir. Bu işlev sayesinde bir tablonun alt kümesi için geçici tablolar yada viewların oluşturulmasının önüne geçilmiş olur. “Where Clause” ifadesi yazıldıktan sonra, “TEST” düğmesine basılarak ifadenin doğruluğu test edilebilir. Test işlemi sırasında girilen ifadenin doğruluğu kontrol edilir ve bu kısıtlamanın kullanılan veriyi ne kadar azalttığı konusunda kullanıcı bilgilendirilir. Eğer ifade doğru değil ise kullanıcı uyarılır. Şekil 4.13 ‘de “where clause” un test edilmesi görülmektedir.

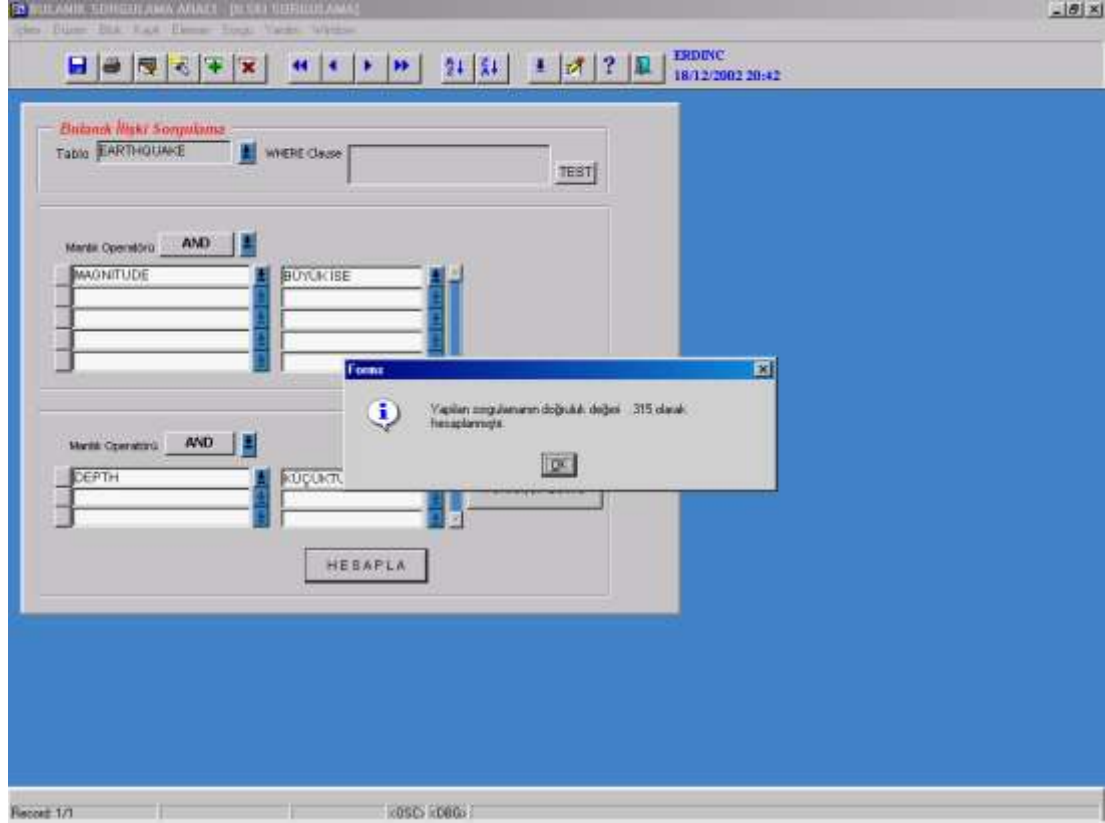


Şekil 4.13: Where ifadesinin test edilmesi

Sonuç bloğundaki her satır için bir bulanık fonksiyon belirlenebilir. Bulanık fonksiyon belirleme işlemi ekrandaki “Fonksiyon Belirle” düğmesi ile yapılır. Açılacak bulanık fonksiyon belirleme penceresi, fonksiyonel ilişki sorgulama ekranındaki gibi çalışır.

Ekrandaki “HESAPLA” düğmesine basılarak belirlenen kuralın doğruluk değeri hesaplanır. Hesaplama işleminin ardından kullanıcı bilgilendirilir, kural ve sonucu veritabanında oluşturulan geçmiş tablolarında saklanır. (Şekil 4.14)

Eğer girilen “Where Clause” doğru yazılıma sahip değil ise hesaplama yapılmaz ve kullanıcı uyarılır.

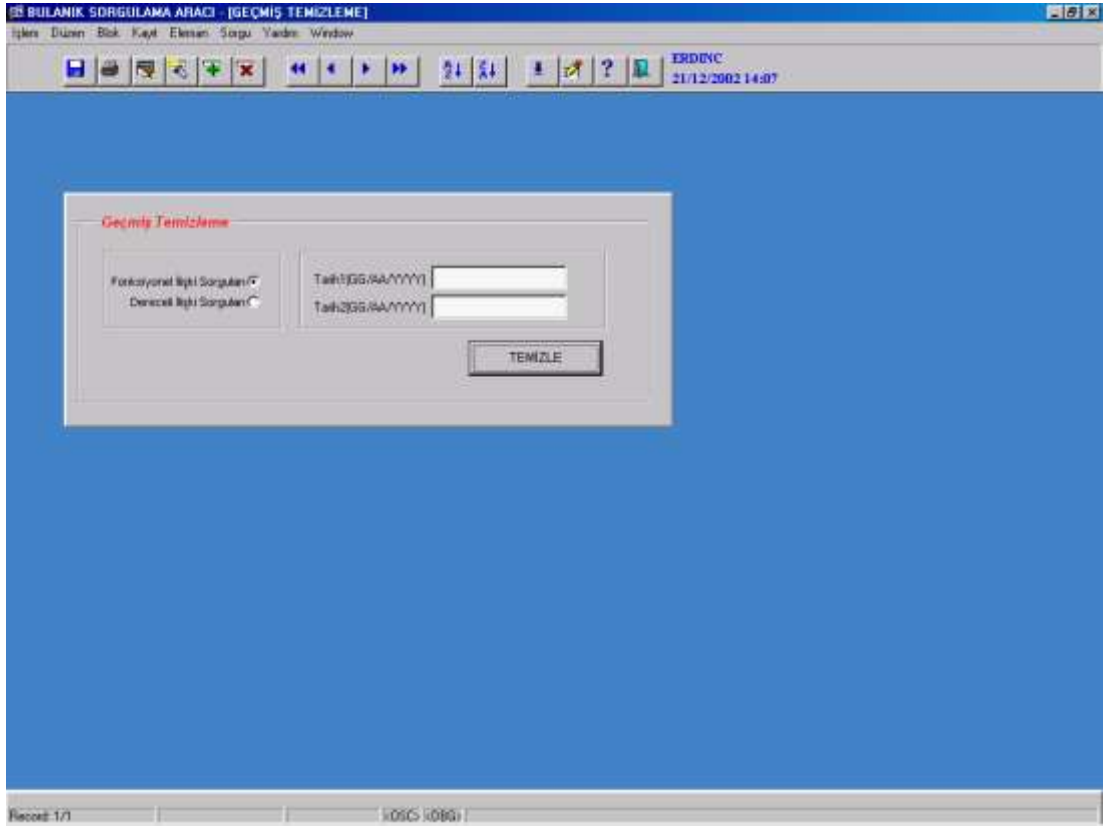


Şekil 4.14: Soru sonucunun görüntülenmesi

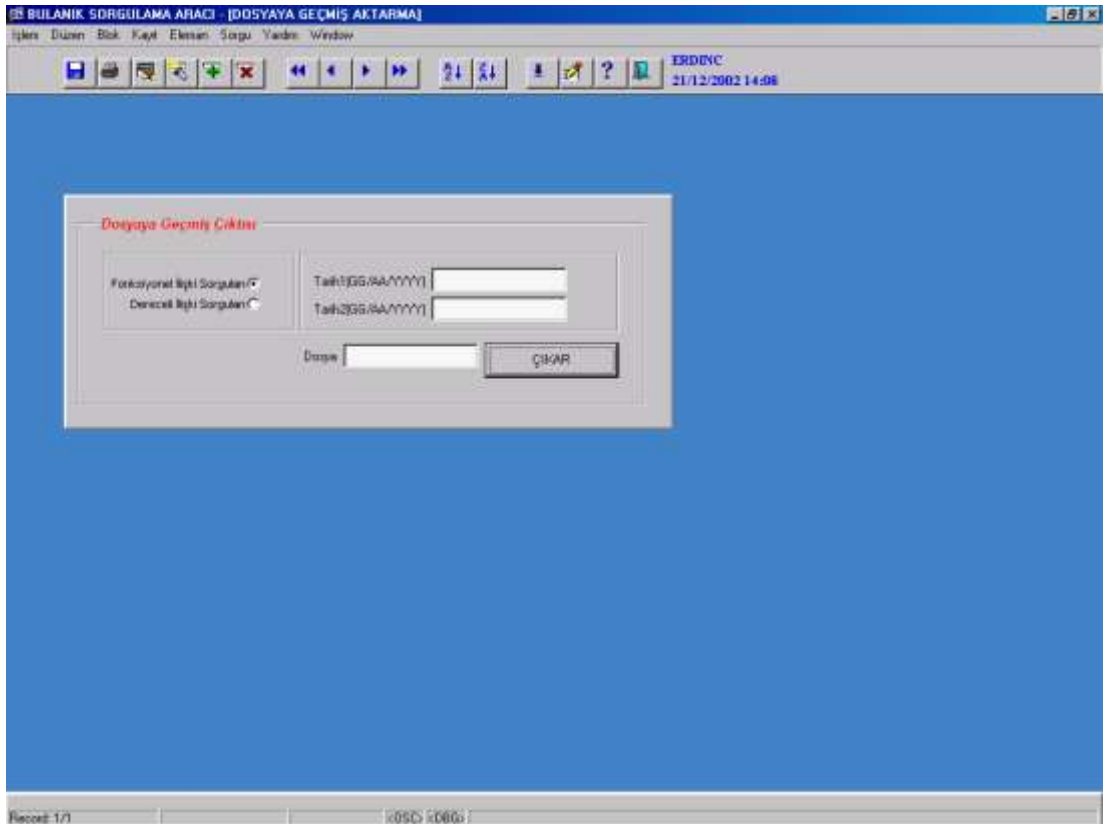
4.5.4 Geçmiş Sorguların Takibi

Geliştirilen araç ile yapılan tüm sorguların işlemleri veritabanında saklanır. Geçmiş sorgu sonuçları üzerinde işlem yapan ekranlara “Geçmiş Sorgular” menüsünden ulaşılabilir. Geçmiş bilgisi sistemi kullanan her kullanıcı için ayrı ayrı tutulur, dolayısıyla bir kullanıcının başka bir kullanıcıya ait geçmiş bilgilerini görmesi mümkün değildir.

“Geçmiş Temizle” ekranı ile istenen tarih aralığındaki geçmiş kayıtlar temizlenir.(Şekil 4.15) Kullanıcı temizleme işlemi öncesi kullanıcıdan onay alınır.

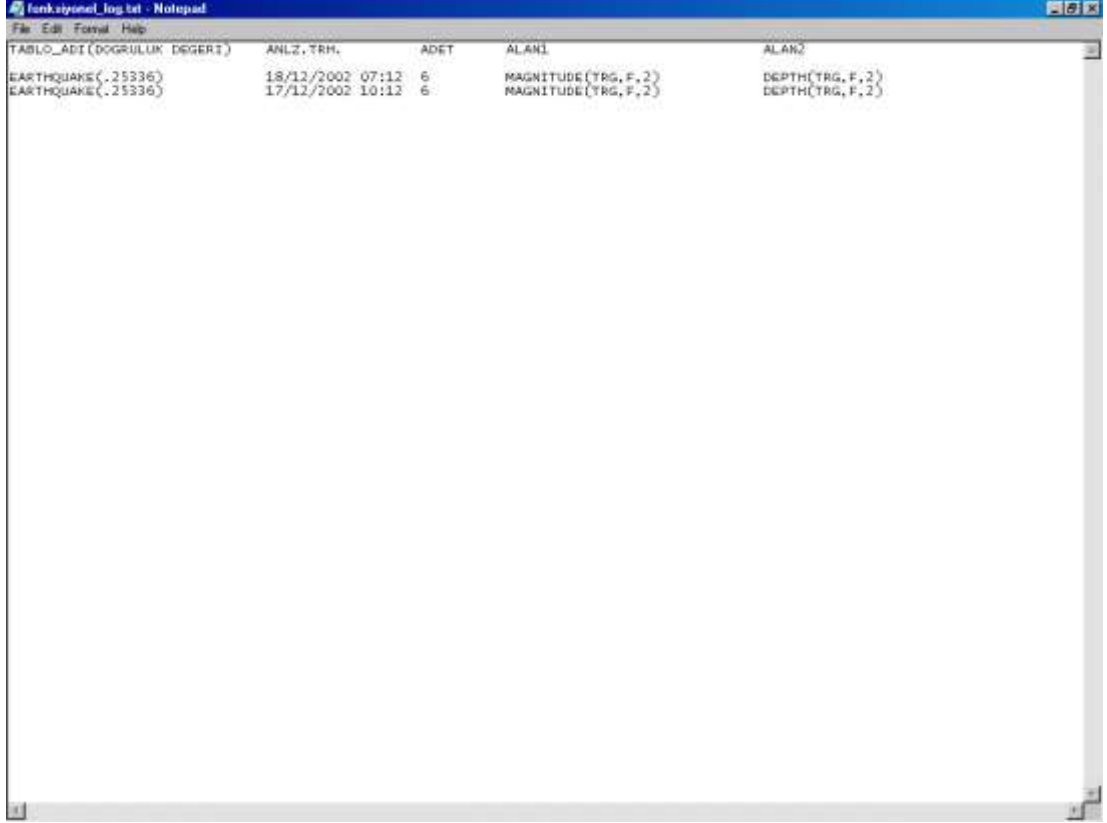


Şekil 4 15: Geçmiş bilgilerin temizlenmesi



Şekil 4 16: Geçmiş bilgilerin dosyaya çıkarılması

Dosyaya geçmiş aktarın ekran ile istenen tarih aralığındaki sorgular bir dosya çıkarılabilir (Şekil 4.16). Gerekli takdirde geliştirilebilecek bir uygulamayla geçmiş veritabanındaki tabloları kullanarak daha etkin raporlama yapılır. Uygulamanın çıkarıldığı dosyalara örnekler Şekil 4.17 ve Şekil 4.18 'de verilmiştir.



The image shows a Notepad window titled 'fonksiyonel_log.txt - Notepad'. The window contains a table with the following data:

TABLO_ADI (DOGRULUK DEGERI)	ANLZ. TRH.	ADET	ALAN1	ALAN2
EARTHQUAKE(.25336)	18/12/2002 07:12	6	MAGNITUDE (TRG, F, 2)	DEPTH (TRG, F, 2)
EARTHQUAKE(.25336)	17/12/2002 10:12	6	MAGNITUDE (TRG, F, 2)	DEPTH (TRG, F, 2)

Şekil 4.17: Fonksiyonel bulan ilişkisi sorgusu için geçmiş dosyası

```
kademek_log.lst - Notepad
File Edit Format Help
TABLO_ADI(DOGRULUK DEGERI) ANLZ.TRH. ADET WHERE_CLAUSE
EARTHQUAKE(.74208) 17/12/2002 10:12 6
  EGER
  ISE MAGNITUDE(TRG,F,2) BUYUK ISE
      DEPTH(TRG,F,2) BUYUKTUR
-----
EARTHQUAKE(.315) 18/12/2002 03:12 6
  EGER
  ISE MAGNITUDE(TRG,F,2) BUYUK ISE
      DEPTH(TRG,F,2) KUCUKTUR
-----
PERSONEL(.86441) 18/12/2002 04:12 20
  EGER
  ISE BOY(TRG,F,2) BUYUK ISE
      VE KILO(TRG,F,2) KUCUK ISE
      YAS(TRG,F,2) BUYUKTUR
-----
PERSONEL(.40066) 18/12/2002 04:12 20
  EGER
  ISE BOY(TRG,F,2) BUYUK ISE
      VE KILO(TRG,F,2) KUCUKTUR
      YAS(TRG,F,2) BUYUKTUR
-----
```

Şekil 4 18: Dereceli bulan k iliş ki sorgusu i ç in geç miş dosyası

5. 2002 YILI TÜRKİYE DEPREMLERİ İLE GERÇEKLEŞTİRİLEN VERİ MADENCİLİĞİ ÇALIŞMASI

Geliştirilen bulanık sorgulama aracı ile Türkiye 'de 2002 yılı içerisinde meydana gelen depremlere ait veriler kullanılarak, bir veri madenciliği çalışması yapılmıştır. Depremlere ait veriler Kandilli Rasathanesi ve Deprem Araştırma Enstitüsü tarafından internet üzerinde yayınlanan “<http://www.koeri.boun.edu.tr/scripts/sondepremler.asp>” sayfasının 13 Ocak 2003 tarihli sürümünden alınmıştır. İlgili deprembilgisi ekteki cd’de “depremler.txt” dosyası içerisinde sunulmuştur.

5.1 Analiz Edilen Verinin Veri Tabanında Saklanması

Veri madenciliği çalışmasında kullanılan veriler Şekil 5.1’de tanımlanan tabloda saklanmıştır. Bu tablodaki “gun” alanı depremin yılın kaçınıcı gününde meydana geldiğini göstermektedir. Çalışma sırasında tarihler arası karşılaştırma yerine günler arası karşılaştırma yapılmıştır.

Veri tabanındaki depremkayıtlarından magnitudü 3.0 değerinden küçük olan 1 adet depremi ihmal edilerek sorgulamalara katılmamıştır.

DEPREM	
TARİH	DATE
GUN	NUMBER(3)
ENLEM	NUMBER(8,4)
BOYLAM	NUMBER(8,4)
DERINLIK	NUMBER(6,4)
MAGNITUD	NUMBER(6,4)
YER	VARCHAR2(50)

Şekil 5.1: Deprem Verilerinin Kaydedildiği Tablo

5.2 Deprem Verisinin Gruplanması

Türkiye’de 2002 yılı içerisinde meydana gelen depremler, coğrafi olarak 6, ve her coğrafi bölgede meydana gelen depremler de, magnitudlarına göre ise 2 adet gruba

ayrılmıştır. Böylece depremler 12 adet .alt deprem grubuna ayrılarak analiz edilmişlerdir.

Sorgulamalarda yapılan bulanık karşılaştırma işlemlerinde tüm alanlar için 4.7 ve 4.13 'de belirtilen bulanık fonksiyon kullanılmıştır. Alanlar için kullanılan eşik değerleri Tablo 5.1 'de verilmiştir.

Tablo 5.1: Alanlar için kullanılan eşik değerleri

ALAN	EŞİK
Derinlik	0.4
Magnitud	0.2
Boylam	1

5.3 Doğruluk Değeri Sorgulanan Önermeler

Sorgulanan önermeler ve elde edilen doğruluk değerleri aşağıdaki gibidir.

39. Paralelin Kuzeyi (enlem ≥ 39)

Magnitüd ≥ 4 (deprem sayısı = 19)

Tablo 5.2: 39. Paralelin kuzeyinde meydana gelen magnitüd > 4.0 dan depremler

ÖNERME	DOĐRULUK
Tarih büyük ise boylam büyüktür / küçüktür.	0.624 / 0.437
Boylam büyük ise magnitüd büyüktür / küçüktür.	0.681 / 0.602
Enlem büyük ise magnitüd büyüktür / küçüktür.	0.681 / 0.602
Boylam büyük ise derinlik büyüktür / küçüktür.	0.395 / 0.724
Enlem büyük ise derinlik büyüktür / küçüktür.	0.623 / 0.496
Derinlik büyük ise magnitüd büyüktür / küçüktür.	0.685 / 0.604
Tarih büyük ise magnitüd büyüktür / küçüktür.	0.636 / 0.650
Tarih büyük ise derinlik büyüktür / küçüktür.	0.331 / 0.783

Magnitüd ≥ 3.0 ve Magnitüd < 4.0 (deprem sayısı = 326)

Tablo 5.3: 39. Paralelin kuzeyinde meydana gelen magnitüd $\in [3.0, 4.0)$ dan depremler

ÖNERME	DOĐRULUK
Tarih büyük ise boylam büyüktür / küçüktür.	0.569 / 0.496
Boylam büyük ise magnitüd büyüktür / küçüktür.	0.704 / 0.568
Enlem büyük ise magnitüd büyüktür / küçüktür.	0.599 / 0.674
Boylam büyük ise derinlik büyüktür / küçüktür.	0.506 / 0.656
Enlem büyük ise derinlik büyüktür / küçüktür.	0.606 / 0.557
Derinlik büyük ise magnitüd büyüktür / küçüktür.	0.601 / 0.673
Tarih büyük ise magnitüd büyüktür / küçüktür.	0.633 / 0.639
Tarih büyük ise derinlik büyüktür / küçüktür.	0.531 / 0.631

39. Paralelin Kuzeyi ve 32. Enlemi'nin Doğusu (enlem ≥ 39 ve boylam ≥ 32)

Magnitüd ≥ 4 (deprem sayısı = 13)

Tablo 5.4: 39. Paralelin kuzeyinde ve 32. Enlemi'nin doğusunda meydana gelen magnitüd > 4.0 dan depremler

ÖNERME	DOĐRULUK
Tarih büyük ise boylam büyüktür / küçüktür.	0.645 / 0.430
Boylam büyük ise magnitüd büyüktür / küçüktür.	0.808 / 0.494
Enlem büyük ise magnitüd büyüktür / küçüktür.	0.571 / 0.731
Boylam büyük ise derinlik büyüktür / küçüktür.	0.622 / 0.615
Enlem büyük ise derinlik büyüktür / küçüktür.	0.673 / 0.564
Derinlik büyük ise magnitüd büyüktür / küçüktür.	0.661 / 0.656
Tarih büyük ise magnitüd büyüktür / küçüktür.	0.760 / 0.546
Tarih büyük ise derinlik büyüktür / küçüktür.	0.398 / 0.832

Magnitüd ≥ 3.0 ve Magnitüd < 4.0 (deprem sayısı = 175)

Tablo 5.5: 39. Paralelin kuzeyinde ve 32. Enlemi'nin doğusunda meydana gelen magnitüd $\in [3.0, 4.0)$ dan depremler

ÖNERME	DOĐRULUK
Tarih büyük ise boylam büyüktür / küçüktür.	0.573 / 0.530
Boylam büyük ise magnitüd büyüktür / küçüktür.	0.682 / 0.578
Enlem büyük ise magnitüd büyüktür / küçüktür.	0.617 / 0.643
Boylam büyük ise derinlik büyüktür / küçüktür.	0.675 / 0.623
Enlem büyük ise derinlik büyüktür / küçüktür.	0.673 / 0.625
Derinlik büyük ise magnitüd büyüktür / küçüktür.	0.621 / 0.638
Tarih büyük ise magnitüd büyüktür / küçüktür.	0.607 / 0.653
Tarih büyük ise derinlik büyüktür / küçüktür.	0.629 / 0.666

39. Paralelin Kuzeyi ve 32. Enlemin Batısı (enlem ≥ 39 ve boylam < 32)

Magnitud ≥ 4 (deprem sayısı = 6)

Tablo 5.6: 39. Paralelin kuzeyinde ve 32. Enlemin batısında meydana gelen magnitud > 4.0 dan depremler

ÖNERME	DOĐRULUK
Tarih büyük ise boylam büyüktür / küçüktür.	0.819 / 0.476
Boylam büyük ise magnitud büyüktür / küçüktür.	0.567 / 0.633
Enlem büyük ise magnitud büyüktür / küçüktür.	0.867 / 0.333
Boylam büyük ise derinlik büyüktür / küçüktür.	0.200 / 0.800
Enlem büyük ise derinlik büyüktür / küçüktür.	0.200 / 0.800
Derinlik büyük ise magnitud büyüktür / küçüktür.	0.700 / 0.500
Tarih büyük ise magnitud büyüktür / küçüktür.	0.300 / 0.900
Tarih büyük ise derinlik büyüktür / küçüktür.	0.267 / 0.733

Magnitud ≥ 3.0 ve Magnitud < 4.0 (deprem sayısı = 151)

Tablo 5.7: 39. Paralelin kuzeyinde ve 32. Enlemin doğusunda meydana gelen magnitud $\in [3.0, 4.0)$ dan depremler

ÖNERME	DOĐRULUK
Tarih büyük ise boylam büyüktür / küçüktür.	0.534 / 0.632
Boylam büyük ise magnitud büyüktür / küçüktür.	0.627 / 0.695
Enlem büyük ise magnitud büyüktür / küçüktür.	0.623 / 0.699
Boylam büyük ise derinlik büyüktür / küçüktür.	0.517 / 0.548
Enlem büyük ise derinlik büyüktür / küçüktür.	0.532 / 0.533
Derinlik büyük ise magnitud büyüktür / küçüktür.	0.647 / 0.676
Tarih büyük ise magnitud büyüktür / küçüktür.	0.650 / 0.672
Tarih büyük ise derinlik büyüktür / küçüktür.	0.467 / 0.598

39. Paralelin Güneyi (enlem < 39)

Magnitud ≥ 4 (deprem sayısı = 44)

Tablo 5.8: 39. Paralelin güneyinde meydana gelen magnitud > 4.0 dan depremler

ÖNERME	DOĐRULUK
Tarih büyük ise boylam büyüktür / küçüktür.	0.635 / 0.486
Boylam büyük ise magnitud büyüktür / küçüktür.	0.591 / 0.590
Enlem büyük ise magnitud büyüktür / küçüktür.	0.628 / 0.554
Boylam büyük ise derinlik büyüktür / küçüktür.	0.529 / 0.613
Enlem büyük ise derinlik büyüktür / küçüktür.	0.375 / 0.768
Derinlik büyük ise magnitud büyüktür / küçüktür.	0.548 / 0.637
Tarih büyük ise magnitud büyüktür / küçüktür.	0.553 / 0.626
Tarih büyük ise derinlik büyüktür / küçüktür.	0.596 / 0.548

Magnitud ≥ 3.0 ve Magnitud < 4.0 (deprem sayısı = 637)

Tablo 5.9: 39. Paralelin güneyinde meydana gelen magnitud $\in [3.0, 4.0)$ dan depremler

ÖNERME	DOĐRULUK
Tarih büyük ise boylam büyüktür / küçüktür.	0.517 / 0.658
Boylam büyük ise magnitud büyüktür / küçüktür.	0.660 / 0.619
Enlem büyük ise magnitud büyüktür / küçüktür.	0.575 / 0.704
Boylam büyük ise derinlik büyüktür / küçüktür.	0.559 / 0.623
Enlem büyük ise derinlik büyüktür / küçüktür.	0.530 / 0.651
Derinlik büyük ise magnitud büyüktür / küçüktür.	0.665 / 0.611
Tarih büyük ise magnitud büyüktür / küçüktür.	0.687 / 0.592
Tarih büyük ise derinlik büyüktür / küçüktür.	0.584 / 0.597

39. Paralelin Güneyi ve 32. Enlemin Doğusu (enlem < 39 ve boylam \geq 32)

Magnitud \geq 4 (deprem sayısı = 15)

Tablo 5.10: 39. Paralelin güneyinde ve 32. Enlemin doğusunda meydana gelen magnitud > 4.0 dan depremler

ÖNERME	DOĐRULUK
Tarih büyük ise boylam büyüktür / küçüktür.	0.613 / 0.459
Boylam büyük ise magnitud büyüktür / küçüktür.	0.738 / 0.471
Enlem büyük ise magnitud büyüktür / küçüktür.	0.662 / 0.548
Boylam büyük ise derinlik büyüktür / küçüktür.	0.343 / 0.800
Enlem büyük ise derinlik büyüktür / küçüktür.	0.381 / 0.762
Derinlik büyük ise magnitud büyüktür / küçüktür.	0.556 / 0.670
Tarih büyük ise magnitud büyüktür / küçüktür.	0.676 / 0.533
Tarih büyük ise derinlik büyüktür / küçüktür.	0.581 / 0.562

Magnitud ≥ 3.0 ve Magnitud < 4.0 (deprem sayısı = 130)

Tablo 5.11: 39. Paralelin güneyinde ve 32. Enlemin doğusunda meydana gelen magnitud $\in [3.0, 4.0)$ dan depremler

ÖNERME	DOĐRULUK
Tarih büyük ise boylam büyüktür / küçüktür.	0.561 / 0.562
Boylam büyük ise magnitud büyüktür / küçüktür.	0.586 / 0.641
Enlem büyük ise magnitud büyüktür / küçüktür.	0.591 / 0.637
Boylam büyük ise derinlik büyüktür / küçüktür.	0.487 / 0.714
Enlem büyük ise derinlik büyüktür / küçüktür.	0.485 / 0.716
Derinlik büyük ise magnitud büyüktür / küçüktür.	0.663 / 0.566
Tarih büyük ise magnitud büyüktür / küçüktür.	0.636 / 0.592
Tarih büyük ise derinlik büyüktür / küçüktür.	0.594 / 0.606

39. Paralelin Güneyi ve 32. Enlemin Batısı (enlem < 39 ve boylam < 32)

Magnitud ≥ 4 (deprem sayısı = 29)

Tablo 5.12: 39. Paralelin güneyinde ve 32. Enlemin batısında meydana gelen magnitud > 4.0 dan depremler

ÖNERME	DOĐRULUK
Tarih büyük ise boylam büyüktür / küçüktür.	0.596 / 0.669
Boylam büyük ise magnitud büyüktür / küçüktür.	0.613 / 0.557
Enlem büyük ise magnitud büyüktür / küçüktür.	0.621 / 0.549
Boylam büyük ise derinlik büyüktür / küçüktür.	0.461 / 0.671
Enlem büyük ise derinlik büyüktür / küçüktür.	0.414 / 0.719
Derinlik büyük ise magnitud büyüktür / küçüktür.	0.552 / 0.618
Tarih büyük ise magnitud büyüktür / küçüktür.	0.492 / 0.677
Tarih büyük ise derinlik büyüktür / küçüktür.	0.590 / 0.543

Magnitud ≥ 3.0 ve Magnitud < 4.0 (deprem sayısı = 507)

Tablo 5.13: 39. Paralelin güneyinde ve 32. Enlemin doğusunda meydana gelen magnitud $\in [3.0, 4.0)$ dan depremler

ÖNERME	DOĐRULUK
Tarih büyük ise boylam büyüktür / küçüktür.	0.499 / 0.770
Boylam büyük ise magnitud büyüktür / küçüktür.	0.615 / 0.689
Enlem büyük ise magnitud büyüktür / küçüktür.	0.702 / 0.717
Boylam büyük ise derinlik büyüktür / küçüktür.	0.554 / 0.622
Enlem büyük ise derinlik büyüktür / küçüktür.	0.537 / 0.639
Derinlik büyük ise magnitud büyüktür / küçüktür.	0.674 / 0.627
Tarih büyük ise magnitud büyüktür / küçüktür.	0.588 / 0.602
Tarih büyük ise derinlik büyüktür / küçüktür.	0.582 / 0.593

5.4 Veri Madenciliđ Çalışması nın Deđerlendirilmesi

Yapılan çalışmada sonrasında elde edilen doğruluk değeri (T) yüksek olan önermelere ilişkin yorumlar 5.4.1 'de verilmiştir. Seçilen önermeler, kendisi ve tersinin doğruluk değerleri arasında 0.2 'den fazla fark olanlardır. Kendisi ve tersinin doğruluk değerlerinin yakın değerler alması bu önermenin belirsizliğini göstermektedir.

5.4.1 Doğruluk Deđeri Yüksek Olan Önermelere İlişkin Yorumlar

39. Paralelin kuzeyinde meydana gelen magnitudu 4.0 'dan büyük olan depremler, tarih ilerledikçe doğuya kaymışlardır. (T=0.624) (Tablo 5.2)

39. Paralelin kuzeyinde meydana gelen magnitudu 4.0 'dan büyük olan depremler, doğuya ilerledikçe yüzeye yakın hale gelmektedirler. (T=0.724) (Tablo 5.2)

39. Paralelin kuzeyinde meydana gelen magnitudu 4.0 'dan büyük olan depremler, tarih ilerledikçe yüzeye yaklaşmışlardır. (T=0.783) (Tablo 5.2)

39. Paralelin kuzeyinde ve 32. Enlemin doğusunda meydana gelen magnitudu 4.0 'dan büyük olan depremlerin, doğuya ilerledikçe magnitudleri büyümüştür. (T=0.808) (Tablo 5.4)

39. Paralelin kuzeyinde ve 32. Enlemin doğusunda meydana gelen magnitudu 4.0 'dan büyük olan depremlerin, tarih ilerledikçe magnitudleri artmıştır. (T=0.760) (Tablo 5.4)

39. Paralelin kuzeyinde ve 32. Enlemin doğusunda meydana gelen magnitudu 4.0 'dan büyük olan depremler, tarih ilerledikçe yüzeye yaklaşmışlardır. (T=0.832) (Tablo 5.4)

39. Paralelin kuzeyinde ve 32. Enlemin batısında meydana gelen magnitudu 4.0 'dan büyük olan depremler, tarih ilerledikçe doğuya doğru ilerlemişlerdir. (T=0.819) (Tablo 5.4)

39. Paralelin kuzeyinde ve 32. Enlemin batısında meydana gelen magnitudu 4.0 'dan büyük olan depremler, tarih ilerledikçe doğuya doğru ilerlemişlerdir. (T=0.819) (Tablo 5.6)

39. Paralelin kuzeyinde ve 32. Enlemin batısında meydana gelen magnitudu 4.0 'dan büyük olan depremlerin, kuzeye ilerledikçe magnituları büyümüştür. (T=0.867) (Tablo 5.6)

39. Paralelin kuzeyinde ve 32. Enlemin batısında meydana gelen magnitudu 4.0 'dan büyük olan depremlerin, doğuya ilerledikçe yüzeye yakın hale gelmişlerdir. (T=0.800) (Tablo 5.6)

39. Paralelin kuzeyinde ve 32. Enlemin batısında meydana gelen magnitudu 4.0 'dan büyük olan depremlerin, derinliği arttıkça magnituları büyümüştür.. (T=0.700) (Tablo 5.6)

39. Paralelin kuzeyinde ve 32. Enlemin batısında meydana gelen magnitudu 4.0 'dan büyük olan depremler, tarih ilerledikçe magnituları küçülmüştür.. (T=0.900) (Tablo 5.6)

39. Paralelin kuzeyinde ve 32. Enlemin batısında meydana gelen magnitudu 4.0 'dan büyük olan depremler, tarih ilerledikçe yüzeye yakın hale gelmişlerdir. (T=0.733) (Tablo 5.6)

39. Paralelin güneyinde meydana gelen magnitudu 4.0 'dan büyük olan depremler, doğuya ilerledikçe yüzeye yakın hale gelmişlerdir. (T=0.768) (Tablo 5.8)

39. Paralelin güneyinde ve 32. Enlemin doğusunda meydana gelen magnitudu 4.0 'dan büyük olan depremlerin, doğuya ilerledikçe magnituları büyümüştür. (T=0.738) (Tablo 5.10)

39. Paralelin güneyinde ve 32. Enlemin doğusunda meydana gelen magnitudu 4.0 'dan büyük olan depremler, doğuya ilerledikçe yüzeye yakın hale gelmişlerdir. (T=0.800) (Tablo 5.10)

39. Paralelin güneyinde ve 32. Enlemin doğusunda meydana gelen magnitudu 4.0 'dan büyük olan depremler, kuzeye ilerledikçe yüzeye yakın hale gelmişlerdir.(T=0.762) (Tablo 5.10)

39. Paralelin güneyinde ve 32. Enlemin doğusunda meydana gelen magnitudu [3.0,4.0) aralığında olan depremler, doğuya ilerledikçe yüzeye yakın hale gelmişlerdir.(T=0.714) (Tablo 5.11)

39. Paralelin güneyinde ve 32. Enlemin doğusunda meydana gelen magnitudu [3.0,4.0) aralığında olan depremler, kuzeye ilerledikçe yüzeye yakın hale gelmişlerdir.(T=0.716) (Tablo 5.11)

39. Paralelin güneyinde ve 32. Enlemin batısında meydana gelen magnitudu 4.0 ' dan büyük olan depremler, doğuya ilerledikçe yüzeye yakın hale gelmişlerdir. (T=0.671) (Tablo 5.12)

39. Paralelin güneyinde ve 32. Enlemin batısında meydana gelen magnitudu 4.0 ' dan büyük olan depremler, kuzeye ilerledikçe yüzeye yakın hale gelmişlerdir. (T=0.671) (Tablo 5.12)

39. Paralelin güneyinde ve 32. Enlemin batısında meydana gelen magnitudu [3.0, 4.0) aralığında olan depremler, tarih ilerledikçe doğru ilerlemişlerdir.. (T=0.770) (Tablo 5.13)

6. SONUÇLAR VE ÖNERİLER

Bu yüksek lisans tez çalışmasının konusu olan “İlişkisel Veritabanlarında Bulanık İlişki Sorgulama Aracı” uygulamasının geliştirme sürecinde Dan Rasmussen ve Ronald Yager tarafından önerilen algoritma temel alınmıştır.

Kullanılan örnek tablolar üzerinde yapılan sorgulamaların tutarlı sonuçlar döndürdüğü gözlenmiştir.

Hesaplama algoritması iç içe çalışan iki döngüden oluşmaktadır. Bu belirli bir işlemin tablodaki kayıt sayısının karesi adedince tekrar edileceği anlamına taşımaktadır. Bunu getireceği performans sorununun an azı indirilebilmesi için, adı geçen döngülerin hafıza birimi üzerinde yapılması sağlanmıştır. Bu sayede, uzak lokasyonlarda da bulunması mali olan veri tabanı tabloları üzerinde sadece bir kez sorgulama yapılması hedeflenmiştir. Ancak bu yaklaşım sebebiyle, büyük tabloların sorgulanması sırasında hafıza biriminin yeterli kapasiteye sahip olmasına dikkat edilmesi gerekir.

Sorgulanan önermelerin doğruluk değerinin 0.5'e yakın olması belirsizlik durumu olarak değerlendirilebilir. Bunun yanı sıra önermenin kendisinin ve tersinin birbirlerine yakın ve yüksek değerler alması seçilen bulanık fonksiyonun eşik değerinin yüksek oluşunu ve önermenin belirsizliğini gösterecektir; bu durumdaki önermelerin doğruluk değerleri yüksek olsa dahi değerlendirme dışı bırakılmalıdır.

Yapılan sorgulamalardan elde edilen doğruluk değerleri incelendiğinde, birçok sorgulamada doğruluk değerinin olasılık ölçüsü gibi davrandığı gözlenmiştir.

Geliştirilen uygulama çok geniş bir alan üzerinde etkin bir şekilde uygulanabilir. Günümüz dünyasının, veritabanı sistemlerinde takip edilen tüm süreçleri üzerinde, fonksiyonel ve dereceli bulanık ilişki sorgusu yapılabilir. Örnek uygulama alanları olarak meteoroloji, jeofizik finans, spor, pazarlama gösterilebilir.

Uygulama kullanıcının girdiği sorgulara doğruluk değeri hesaplayan ve sonuçlarını veri tabanındaki geçmiş tablolarında saklayan yapıda geliştirilmiştir. Uygulamanın belirli bir tablo üzerinde tanımlanabilecek tüm sorguları otomatik olarak oluşturup doğruluk değerlerini hesaplayan ve raporlayan bir yapıda tekrar düzenlenmesi faydalı

olacaktır. Bu sayede doğruluk değeri yüksek olan fakat analistin sorgulama teşebbüsünde bulunmadığı kurallar da açığa çıkarılmış olacaktır.

Bunun yanı sıra uygulama, yapılan sorguların doğruluk değerlerini, tablo yapısının değiştiği varsayımı ile, her seferinde tekrar hesaplamaktadır. Çok büyük tabloların sorgularının alacağı zaman gözönünde bulundurulduğunda, kullanıcının, benzer geçmiş sorgular konusunda uyarılması yerinde olacaktır.

Geliştirilen aracın kullanarak yapılacak sorgular sayesinde çeşitli veritabanlarından anlamlı sonuçlar çıkarmak mümkün olacaktır. Çıkarılacak sonuçlar, ilgili sistem üzerinde yapılacak geliştirme çalışmalarına ışık kaynağı olabilir.

KAYNAKLAR

- [1] **Rasmussen, D and Yager, R R** , 1999. Finding Fuzzy and Gradual Functional Dependencies with Summary Sql, *Fuzzy Sets and Systems* , **106**, 131-142
- [2] **Rasmussen, D and Yager, R R** , 1997. Flexible Query Answering Systems, Kluwer Academic Publishers, United States of America
- [3] **Kir, G.J. and Folger, T A** , 1988. Fuzzy Sets, Uncertainty and Information , Prentice Hall
- [4] **Akpinar, H** , 2000. Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği, *İ. Ü İşletme Fakültesi Dergisi*, **1**, 1-22
- [5] **Apaydın, E** , 2000. Zeki Veri Madenciliği, Bilişim2000 Eğitim Semineri
- [6] **Yonar, Y B** , 1999. Genel Amaçlı Bir Uzman Sistem Yüksek Lisans Tezi, İ. T. Ü Fen Bilimleri Enstitüsü İstanbul.

EK A

Aşağıda geliştirilen uygulamada içerisinde kullanılan kritik kodlar yer almaktadır.

Fonksiyonel Sorgu Hesaplama Butonu Arkasında Çalışan Kod

```
DECLARE
al_id Alert;
al_button Number;
X NUMBER(24,3);
v_cursor1 INTEGER;
v_cursor2 INTEGER;
v_sart_alan_adet NUMBER(2);
v_sonuc_alan_adet NUMBER(2);
vt_degerler fuzzy.type_number;
vt_alan fuzzy.type_char;
vt_sart_fnk_tipi fuzzy.type_char;
vt_sart_fark_tipi fuzzy.type_char;
vt_sart_esik fuzzy.type_number;
vt_alan_deger1 fuzzy.type_number;
vt_alan_deger2 fuzzy.type_number;
v_sart_mantik_op VARCHAR2(1);
v_sonuc_mantik_op VARCHAR2(1);
i NUMBER;
j NUMBER;
k NUMBER;
v_kayit_sayisi NUMBER;
v_select_list VARCHAR2(500);
v_query_str VARCHAR2(500);
ignore INTEGER;
v_sart NUMBER;
v_sart_ters NUMBER;
v_sart_genel NUMBER;
v_sonuc NUMBER;
v_sonuc_ters NUMBER;
v_sonuc_genel NUMBER;
v_alt_toplama NUMBER;
v_ust_toplama NUMBER;
v_benzerlik1 NUMBER;
v_benzerlik2 NUMBER;
v_genel_toplama NUMBER;

BEGIN
ALANLARI_KONROL_ET;
-- Ekrandan sorgu için girilen alanlar okunuyor
```

```

        GO_BLOCK(' CONTROL ');
        FIRST_RECORD;
        vt_alan(1) := CONTROL.ALAN1;
        vt_alan(2) := CONTROL.ALAN2;
        -- Veri tabani tablosundan okunacak alanlar select_list seklinde hazirlaniyor...
        v_select_list := vt_alan(1) || ',' || vt_alan(2);

        --/ Veri tabani tablosundan okunacak alanlar select_list seklinde hazirlaniyor...

        v_query_str := 'SELECT' || v_select_list || ' FROM' || :CONTROL.TABLO_ADI;
        v_cursor1 := dbms_sql.open_cursor;
        DBMS_SQL.PARSE(v_cursor1, v_query_str, 2);
        FOR i IN 1..2 LOOP
            DBMS_SQL.DEFINE_COLUMN(v_cursor1, i, X);
        END LOOP;
        ignore := DBMS_SQL.EXECUTE(v_cursor1);

        -- 1. LOOP
        i := 1;
        LOOP
            IF DBMS_SQL.FETCH_ROWS(v_cursor1) = 0 THEN
                EXIT;
            ENDIF;
            -- Alanlari n degerleri okunuyor
            FOR j IN 1..2 LOOP
                DBMS_SQL.COLUMN_VALUE(v_cursor1, j, vt_degerler(i));
                i := i + 1;
            END LOOP;
            --/ Alanlari n degerleri okunuyor
        END LOOP;
        DBMS_SQL.CLOSE_CURSOR(v_cursor1); -- Cursor kapatiliyor.
            -- Ilgili tablonun tum satirlari pl/sql table'a alindi.
            -- Tum alanlar ardisik olarak bir pl/sql table yazildi.
            -- Ayristirma gerekecek
        v_kayit_sayisi := (i-1)/2;
        v_genel_toplam := 0;
        <<LOOP_1>>
        FOR i IN 1..v_kayit_sayisi LOOP -- 1. LOOP
            vt_alan_deger1(1) := vt_degerler((i-1)*2+1);
            vt_alan_deger1(2) := vt_degerler((i-1)*2+2);

            v_ust_toplam := 0;
            v_alt_toplam := 0;
            <<LOOP_2>>
            FOR j IN 1..v_kayit_sayisi LOOP -- 2. LOOP
                vt_alan_deger2(1) := vt_degerler((j-1)*2+1);
                vt_alan_deger2(2) := vt_degerler((j-1)*2+2);

                v_benzerlik1 := FUZZYIQ(vt_alan_deger1(1), vt_alan_deger2(1), 'B',
:contrd.fonksiyon_tipi1, :contrd.fark_tipi1, :control.esik1, 0);

```



```

        v_benzerlik2 := FUZZY.IQ(vt_alan_deger1(2), vt_alan_deger2(2), 'B,
:contrd.fonksiyon_tipi2:contrd.fark_tipi2:control.esik2, 0);
        v_ust_toplam := v_ust_toplam + LEAST(v_benzerlik1, v_benzerlik2);
        v_alt_toplam := v_alt_toplam + v_benzerlik1;
    END LOOP;
    v_genel_toplam := v_genel_toplam + (v_ust_toplam / v_alt_toplam);
END LOOP;
LOG_A_YAZ(v_genel_toplam / v_kayit_sayisi);
al_id := Find_Alert(' MESA');
Set_Alert_Property(al_id, alert_message_text, 'Yapilan sorgulamanın
doğruluk değeri ' || TO_CHAR(v_genel_toplam / v_kayit_sayisi, '9D999') || ' olarak
hesaplanmıştır. ');
al_button := Show_Alert(al_id);
END;

```

Fuzzy Paketi

```

PACKAGE BODY FUZZY IS

```

```

--İliski Operatörü

```

```

FUNCTION IQ(v1 NUMBER, v2 NUMBER, v_op CHAR, v_fnk CHAR,
v_f_tipi CHAR, v_esik NUMBER, v_degil NUMBER) RETURN NUMBER IS

```

```

    v_deger NUMBER;
    v_kucuk NUMBER;
    v_buyuk NUMBER;
    v_fark_deger NUMBER;
    v_op2 VARCHAR2(2);

```

```

BEGIN

```

```

    IF v1 < v2 THEN
        v_kucuk := 1;
        v_buyuk := 0;
    ELSIF v1 > v2 THEN
        v_kucuk := 0;
        v_buyuk := 1;
    ELSE
        v_kucuk := 0;
        v_buyuk := 0;
    END IF;

```

```

    IF v_degil = 1 THEN
        IF v_op = 'BB' THEN
            v_op2 := 'KB'; END IF;
        IF v_op = 'KB' THEN
            v_op2 := 'BB'; END IF;
        IF v_op = 'KU' THEN
            v_op2 := 'BU'; END IF;
        IF v_op = 'BU' THEN
            v_op2 := 'KU'; END IF;
    ELSE
        v_op2 := v_op;
    END IF;

```

```

IF v_f_tipi = ' F ' THEN
  v_fark_deger := ABS(v1-v2);
ELSIF v_f_tipi = ' O ' THEN
  v_fark_deger := LEAST(v1, v2)/ GREATEST(v1, v2);
ENDIF;

IF v_op2 = ' B ' THEN
  IF v_fnk = ' TRG ' THEN
    IF v_f_tipi = ' F ' THEN
      IF v_fark_deger < v_esi k THEN
        v_deger := (v_esi k-v_fark_deger)/v_esi k;
      ELSE
        v_deger := 0;
      ENDIF;
    ELSIF v_f_tipi = ' O ' THEN
      IF v_fark_deger > v_esi k THEN
        v_deger := ((1-v_esi k)-(1-v_fark_deger))/(1-v_esi k);
      ELSE
        v_deger := 0;
      ENDIF;
    ENDIF;
  ELSIF v_fnk = ' DPRB ' THEN
    IF v_f_tipi = ' F ' THEN
      IF v_fark_deger < v_esi k THEN
        v_deger := POWER(v_esi k-v_fark_deger, 2)/POWER(v_esi k, 2);
      ELSE
        v_deger := 0;
      ENDIF;
    ELSIF v_f_tipi = ' O ' THEN
      IF v_fark_deger > v_esi k THEN
        v_deger := POWER((1-v_esi k)-(1-v_fark_deger), 2)/POWER(1-v_esi k, 2);
      ELSE
        v_deger := 0;
      ENDIF;
    ENDIF;
  ELSIF v_fnk = ' YPRB ' THEN
    IF v_f_tipi = ' F ' THEN
      IF v_fark_deger < v_esi k THEN
        v_deger := POWER(v_esi k-v_fark_deger, 0.5)/POWER(v_esi k, 0.5);
      ELSE
        v_deger := 0;
      ENDIF;
    ELSIF v_f_tipi = ' O ' THEN
      IF v_fark_deger > v_esi k THEN
        v_deger := POWER((1-v_esi k)-(1-v_fark_deger), 0.5)/POWER(1-
v_esi k, 0.5);
      ELSE
        v_deger := 0;
      ENDIF;
    ENDIF;
  ENDIF;
ENDIF;

```

```

        ENDIF;
    ENDIF;
ENDIF;
RETURN(v_deger);
END;

/*-----*/
FUNCTION MQ(v_adet NUMBER, vt_deger1 type_number, v_m_op CHAR,
v_degi1 NUMBER) RETURN NUMBER IS
    v_deger NUMBER;
BEGIN
    IF v_m_op = ' AND ' THEN
        v_deger := 1;
        FOR i IN 1..v_adet LOOP
            v_deger := LEAST(v_deger, vt_deger1(i));
        END LOOP;
    ELSE
        v_deger := 0;
        FOR i IN 1..v_adet LOOP
            v_deger := GREATEST(v_deger, vt_deger1(i));
        END LOOP;
    END IF;

    IF v_degi1 = 1 THEN
        RETURN(1-v_deger);
    ELSE
        RETURN(v_deger);
    END IF;
END;

/*-----*/

END;

```

ÖZGEÇMİŞ

Erdoğan Fadıl ERTANİŞ 1975 yılında İstanbul'da doğdu. 1993 yılında İstanbul Özel Fatih Erkek Lisesi'nden mezun oldu. Aynı yıl, İstanbul Teknik Üniversitesi Fen Edebiyat Fakültesi, Matematik Mühendisliği bölümünde lisans öğrenimine başladı. Matematik Mühendisliği bölümünden 1997 yılında mezun oldu, aynı yıl İstanbul Teknik Üniversitesi, Sistem Analizi programında yüksek lisans öğrenimine başladı. Halen, Koç Sistem Bilgi ve İletişim Hizmetleri A.Ş.'de uzman olarak görev yapmaktadır.