

T.C  
ÇANAKKALE ONSEKİZMART ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

139069

ÜNİVERSİTE BİLGİ YÖNETİM SİSTEMİ İÇİN  
VERİ AMBARI VE VERİ MADENCİLİĞİNİN  
OLUŞTURULMASI

YÜKSEK LİSANS TEZİ

139069

Hazırlayan : Yasemin YÜKSEK  
Danışman : Doç.Dr. M.Ali SALAHLI

ÇANAKKALE-2003

Çanakkale Onsekiz Mart Üniversitesi Fen Bilimleri Enstitüsü Müdürlüğüne,

Bu araştırma jürimiz tarafından Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans Tezi olarak kabul edilmiştir.

Başkan : Yrd. Doc. Dr. Servet SENYÜCEL

Üye : Doc. Dr. M. Ali SALAHLI

Üye : Doc. Dr. Yakup HACIYEV

Üye :

Üye :

Kod No: 89

Yukarıdaki imzaların adı geçen öğretim üyelerine ait olduğunu onaylıyorum.



## İÇİNDEKİLER

ÖZ.....	I
ABSTRACT .....	II
SİMGELER VE KISALTMALAR.....	III
ÇİZELGELER .....	V
ŞEKİLLER.....	VI
GİRİŞ.....	1
1. GELENEKSEL VERİ YÖNETİMİ SİSTEMLERİ VE ONLARIN UYGULAMA YETERSİZLİKLERİ .....	3
1.1 Geleneksel Veri Yönetimi.....	3
1.2 Veritabanı Yönetim Sisteminde Veri Modelleri.....	3
1.3 İlişkisel Veritabanı Yönetim Sistemleri.....	4
1.4 Varlık Bağını Modelinde Genelleme ve Kümeleme Örneği .....	5
1.5 Nesneye Yönelik Veritabanı Yönetim Sistemleri.....	6
1.6 Dağıtık Veritabanı Yönetim Sistemleri.....	8
1.7 Veritabanı Yönetim Sistemlerinin İş Uygulamalarında Yetersizlikleri ve Veri Yönetiminde Çağdaş Gelişmeler.....	10
1.8 Çokboyutsal Veritabanları.....	11
2. VERİ AMBARLARI TANIMI, GELİŞİMİ VE TASARIMI.....	12
2.1 Veri Ambarı Tanımı.....	12
2.2 Veri Ambarlama Kavramları.....	12
2.3 Veri Ambarı Oluşturma ve Gelişimi.....	13
2.4 Veri Ambarı Mimarisi.....	16
2.5 Veri Ambarlama Teknolojileri.....	17
2.6 WWW’de Veri Ambarlama.....	27
2.7 Mevcut Veritabanları ile Veri Ambarı Tasarımı.....	28
3. VERİ MADENCİLİĞİ.....	29
3.1 Veri Madenciliği Tanımı.....	29
3.2 Veri Madenciliğinde İzlenen Adımlar.....	29
3.3 Veri Madenciliği İşlemleri.....	34
3.4 Veri Madenciliği Teknolojileri.....	36

---

3.5 Veri Madenciliği Yöntemleri.....	37
3.5.1 Sınıflama.....	37
3.5.2 Kümeleme.....	39
3.5.3 Birliktelik Kuralları ve Ardaşık Zamanlı Örüntüler.....	40
3.5.4 Karar Analizi.....	41
3.6 Öğrenmedeki Modellerin Kurulması .....	42
3.6.1 Denetimli Öğrenme.....	43
3.6.2 Denetimsiz Öğrenme.....	43
3.7 Veri Madenciliğindeki Problemler.....	44
3.8 Veri Madenciliği Üzerine Yapılan Çalışmalar.....	44
3.9 Mevcut Veritabanlarında Analiz Servisleri ile Veri Hazırlama ve Veri Madenciliği.....	47
3.10 WWW’te Veri Madenciliği.....	48
4.ÜNİVERSİTE BİLGİ SİSTEMİ İÇİN VERİ AMBARININ OLUŞTURULMASI İLKELERİ VE ÖRNEK UYGULAMALAR.....	50
4.1 Problemin Tanımı .....	50
4.2 Şemaların Tasarımı.....	53
4.3 Raporların Oluşturulması.....	56
SONUÇ.....	71
ÖZET.....	73
SUMMARY.....	75
KAYNAKLAR.....	77
TEŞEKKÜR.....	80
ÖZGEÇMİŞ.....	81

## ÖZ

Yöneticiler için karar vermede bilgi ve verilerin önemi hızla artmaktadır. Bunun başlıca nedenleri rekabet ortamının oluşumu, küreselleşme, iletişim teknolojilerindeki hızlı gelişmelerdir. Böyle bir ortamda gereken verileri elde edebilme, veriye ve bilgiye hızlı erişebilme, ortak veri kümesi içinden doğru bilgileri zamanında alabilmek kurumların ayakta kalmasını ve rekabet gücü oluşturabilmesini sağlayacak bileşenlerdir.

Karar vermek için gerekli bilgiler kurum içinde veya kurum dışında farklı kaynaklardan alınabilmektedir. Veri saklama ve erişim modelleri farklı olabilmektedir. Son kullanıcılar karar vermek için gerekli bilgi ve verilere doğrudan erişim sağlayamamaktadır. Bu veriler, belirlenmiş algoritmaların yardımıyla sağlanmaktadır.

Karar verme ve yöneticiler için verilerin ayrıntılardan arındırılmış ve belirlenmiş algoritmalar ve yöntemlerle gereken doğru bilgilerin elde edilmesi veri ambarcılığı ve veri madenciliğinin temel amaçlarıdır.

Tezde veri ambarcılığı ve veri madenciliği problemleri araştırılarak gelişen yönleri incelenmiştir. Ve Üniversite Bilgi Sisteminde veri ambarı ve veri madenciliği uygulaması yönleri incelenmiştir. Üniversite veri ambarının mimarisi önerilmiştir. Ve uygulama örnekleri verilmiştir.

*Anahtar Kelimeler:* Veri Ambarı, Çevrimiçi Analitik İşlem İşleme(OLAP), Yıldız Şema, Veri Madenciliği, Üniversite Veri Ambarı.

## **ABSTRACT**

For manager's deciding, information and datas' importance increases more by more. The main reasons of it, is competiton condition. It means globalization and improvements on communication technologies fastly.

In that situation, obtaining necessary data reaching datas and informations more quickly, taking the true datas from common data set at the true times are components of which provide companies stand by and give competition power.

For deciding the necessary informations can be taken from resources of inside of company or bised of company. Models of data saving or data reaching are different. End users can not reach the true and necessary information and datas directly. This datas and and informations become for defined algorithmys.

The main purposes of data warehouse and data mining is that removing details of defining algorithms and taking necessary true informations by methods.

In thesis, the improving features of data mining's and data warehouse's problems were searched. Also in information system of the university, data warehouses and data mining application's alternatives were studied. The data warehouse structure of univeristy was suggested and application examples were given.

**Key Words :** Data Warehouse, Online Analytical Processing(OLAP), Star Scheme, Data Mining, University Data Warehouse.

## SİMGELER VE KISALTMALAR

VTYS	Veritabanı Yönetim Sistemi
OLAP	Çevrimiçi Analitik İşleme
KDS	Karar Destek Sistemleri
OLTP	Çevrimiçi Hareket İşlemleri
ODS	İşletimsel Veri Sistemleri
YBS	Yönetici Bilgi Sistemleri
ROLAP	İlişkisel İşletimsel Veri Sistemleri
MOLAP	Çokboyutsal İşletimsel Veri Sistemleri
SQL	Standart Query Language
ISA	Is a Kind Of
NYVM	Nesneye Yönelik Veri Modeli
NYVT	Nesneye Yönelik Veritabanı
NYVTYS	Nesneye Yönelik Veritabanı Yönetim Sistemi
DVYT	Dağıtık Veritabanı
DVTYS	Dağıtık Veritabanı Yönetim Sistemi
DQP	Dağıtık Sorgu İşleme
DTM	Dağıtık Dönüştürme Yönetimi
DMM	Dağıtık Metadata Yönetimi
DSP	Dağıtık Güvenlik Yönetimi
DIM	Dağıtık Bütünleşik Yönetimi
VM	Veri Madenciliği
VA	Veri Ambarı
VAM	Veri Ambarı Mimarisi
WWW	World Wide Web
XML	Extensible Markup Language(Genişleyebilir İşaretleme Dili)
HTML	HyperText Markup Language(Hareketli-Metin İşaretleme Dili)
MDX	Çokboyutsal Anlatım
DTS	Veri Değişim Servisleri
HOLAP	Hibrid OLAP(Melez Çevrimiçi Analitik İşleme)
İVTYS	İlişkisel Veritabanı Yönetim Sistemi
ODBC	Online Database Connectivity

<b>ÇBVTYS</b>	<b>Çokboyutsal Veritabanı Yönetim Sistemi</b>
<b>MÖ</b>	<b>Makina Öğrenme</b>
<b>OLAM</b>	<b>Çevrimiçi Analitik Madencilik</b>
<b>OLAP</b>	<b>Çevrimiçi Analitik İşleme</b>
<b>DMQL</b>	<b>Veri Madenciliği Sorgulama Dili</b>
<b>ERP</b>	<b>Kurumsal Kaynak Planlama</b>
<b>CRM</b>	<b>Customer Relationship Management(Müşteri İlişki Yönetimi )</b>
<b>PBS</b>	<b>Personel Bilgi Sistemi</b>





## ÇİZELGELER

Çizelge No	Çizelge Adı	Sayfa No
Çizelge 1.1	OLTP Sistemleri ve OLAP Sistemleri karşılaştırılması	9
Çizelge 2.1	Birimler ve Zaman boyutuna ait toplam sayı tablosu	21
Çizelge 2.2	Verilerin iki boyutlu matris şeklinde gösterimi	23
Çizelge 2.3	Toplam sayının Roll-Up gösterimi	24
Çizelge 2.4	GROUP BY gösterimi	25
Çizelge 2.5	WITH ROLLUP gösterimi	26
Çizelge 2.6	CUBE gösterimi	26
Çizelge 3.1	VM'nin evrimsel gelişim adımları	31
Çizelge 3.2	Gelir-birikim verileri	33
Çizelge 3.3	Gelir-birikim verilerin ortalama tablosu	33
Çizelge 4.1	Çapraz tablo gösterimi	62

## ŞEKİLLER

Şekil No	Şekil Adı	Sayfa No
Şekil 1.1	Genelleme örneği	5
Şekil 1.2	Kümeleme örneği	6
Şekil 1.3	Nesneye Yönelik Veri Modeli esasları	6
Şekil 2.1	Veri Ambarının genel görünümü	13
Şekil 2.2	Yıldız Şema Gösterimi	18
Şekil 2.3	Normalize Edilmiş Yıldız Şeması	21
Şekil 2.4	Verilerin 3 boyutlu şeklinde gösterimi	23
Şekil 3.1	Veritabanı Bilgi Keşfetme adımları	31
Şekil 3.2	Regresyon Denkleminin grafik gösterimi	34
Şekil 3.3	Yapay Sinir Ağları modeli	38
Şekil 3.4	Karar Ağacı	42
Şekil 3.5	DBMiner Mimarisi	45
Şekil 3.6	Web Madenciliği sınıflandırılması	48
Şekil 4.1	Üniversite Veri Ambarının genel yapısı	51
Şekil 4.2	PBS'nin Mantıksal Modeli	53
Şekil 4.3	Personel Veri Tabanının bir bölümü üzerindeki tablolara ait ilişkiler	55
Şekil 4.4	Boyutsal Hiyerarşi	55
Şekil 4.5	Birim, ünvan ve yıl parametrelerinin koordinat ekseninde gösterimi	56
Şekil 4.6	Rapor Seçimi	58
Şekil 4.7	Akademik Ünvana göre öğretim elemanı kadro sayısı	59
Şekil 4.8	Fakülte-Birimler içerisinde öğrenci sayısı	63

<b>Şekil No</b>	<b>Şekil Adı</b>	<b>Sayfa No</b>
Şekil 4.9	Giriş Puanların-Taban puanların listelenmesi	64
Şekil 4.10	Mühendislik-Mimarlık fakültesine ait mezuniyet puan listesi	65
Şekil 4.11	Makale sayısı liste durumu	66
Şekil 4.12	Yıllara göre herbir fakültede öğrenci sayısı	67
Şekil 4.13	Yeni tablo oluşturma	69
Şekil 4.14	Yeni rapor oluşturma	70



## GİRİŞ

Teknolojinin ilerlemesiyle son yıllarda veri üretimi ve kullanımı artmaktadır. Veriler üzerinde analitik işlemler kullanılarak sistemler tarafından istenilen bilgiler elde edilebilmektedir. Bu bilgilere ulaşılmasında çağdaş teknolojiler kullanılmaktadır. Bu uygulamada veri ambarı, veri madenciliği geliştirilmiştir.

İşletimsel kaynaklardan sonuçlar çıkararak karar desteğe yönelik bilgiler toplanması çok zor ve güç olup çoğu zaman istenen sonuca ulaşamayan çalışmalar gözlenilmektedir. Değişik yapıda birbirinden uzak ve çok büyük bilgi kaynaklarında bu güçlük daha da artmaktadır. Bu zorluklarını yenmek için VA'ları mimarisi oluşturulmuştur.

Veri Ambarı oluşumunda veriler daha çok değişik kaynaklarda bulunması, değişik görümlere ve yapılaraya sahip olması nedeniyle Karar Destek Sistem(KDS)'leri gibi sorgu araçları, Çevrimiçi-Analitik İşleme(OLAP) analizleri kullanılarak verilerin tek şema altında toplanmasını sağlanılmıştır.

90'lı yılların başında W.H. Inmon tarafından ileri sürülmüş, veri ambarı bu anlamda güçlü bir mimaridir. Veri Ambarındaki veri, karar desteğe yönelik olarak depolanmaktadır. Farklı kaynaklardan oluşan sistemlerde verinin ortak uygulamaya yönelik olarak bütünleşik veri kaynakları, kullanıcıya verinin birleşmiş şeklini sunmaktadır. Ve ambardaki veri gerçek zamanda güncelleşmeyen fakat işletimsel sistemlerde düzenli olarak yenilenebilir.

Veri Ambarları ve veritabanlarında çok büyük sayılarda veri depolanmaktadır. Bu veri yığınlarından veri çıkarımında kullanıcıya kolaylık sağlayan uygulamalardan biri olarak veri madenciliği sunulmuştur. Veri Madenciliği yöntemleri olarak sınıflama, kümeleme, birliktelik kuralları, ardaşık örüntüler ve regresyon kuralları vardır.

Tez konusunun amacı veri ambarının oluşturulması ve veri ambarları üzerinde veri çıkarımının bir adımı olan veri madenciliği üzerine araştırma yapmaktır. Bu amaçla;

- Veri modellerinin ve veri tabanlarının tarihsel gelişimi incelenmiş, ilişkisel veri modelleri, onların başarılı ve yetersiz yönleri araştırılmıştır.
- Bilgilerin işlenmesinde yeni bir gelişim aşaması olan veri ambarı oluşturmada yıldız şema türleri ve veri küpleri incelenmiştir.
- VT' larındaki zaman boyutu olmayan, güncel, değişebilen, ayrıntılı

verilerden karar desteęe yönelik bilgiler toplanmasını kolaylařtırmak ve çoęu zaman istenen sonuca ulařılamayan alıřmalar sonucundaki bu gclk ve zorluklarını yenmek iin VA'ları mimari yapısı incelenmiřtir.

- Veri Ambarı mimarisi zerindeki temel kavramlar tez'de verilmiřtir.
- Veri Ambarlarında verilerin tasviri ve iřlenilmesi yntemleri

arařtırılmıřtır

• İřlenmemiř veriden gerekli bilgi ıkarma teknolojilerinden biri olan veri madencilięi adımları incelenmiřtir.

• Veri Madencilięi yntemleri, teknolojileri ve iřlemleri tez'de incelenmiřtir.

• niversite Bilgi Sistemi iin Veri Ambarının mimarisi nerilmiř ve rnek uygulamalar yapılmıřtır.



# 1.GELENEKSEL VERİ YÖNETİMİ SİSTEMLERİ VE ONLARIN UYGULAMA YETERSİZLİKLERİ

## 1.1 Geleneksel Veri Yönetimleri

Veritabanı, çeşitli kullanıcılar tarafından çok çeşitli amaçlarla kullanılmak üzere verilerin bir araya getirilmesi sonucunda ortaya çıkan veri kümeleri olarak tanımlanabilir.

Veritabanı Yönetim Sistemi kullanıcıya ve işletmeye aşağıda görülen avantajları sağlayabilecektir:

1. Veri tabanındaki veriler gereksiz yinelenmelerden arınmış olarak, düzenli bir biçimde bilgisayar belleklerinde saklanır.
2. Verilerin doğruluğunun ve bütünlüğünün arttırılmasıyla birçok uygulama ilişkili ve işletimsel verilerden oluşur.
3. Kullanıcının verileri kullanabilme seviyesini arttırmaktadır. Fakat verilere doğrudan erişemezler. VT, VTYS yazılımlarıyla merkezi denetimi sağlar.
4. Verilerin gizliliğinin korunması, kullanıcıların yetkili oldukları işlemleri yapabilmesiyle daha iyi kontrol sağlanır.

## 1.2 Veritabanı Yönetim Sisteminde Veri Modelleri

Her Veri Tabanı Yönetim Sistemde veri modeli kullanılır. Veri modelleri geleneksel ve anlamsal modeller olarak iki ana alanda toplanılmaktadır. Geleneksel modeller, ağ, hiyerarşi ve ilişkisel modellerdir. Anlamsal modeller ise, varlık-bağıntı, fonksiyonel, mantıksal tabanlı ve nesneye yönelik modeller örnek verilebilir. Modeller ile birlikte VTYS'lerinin gelişmesi üç kısımda incelenebilir.

İlk kısımda 1960 ve 1970'li yıllarda sıradüzensel ve ağ modelleri kullanılmıştır. Bu modellerin VTYS'ne uyguladığı yenilikler fiziksel-mantıksal veri modeli oluşturma, veri tanımlarının sürekliliği ve fiziksel veri bağımsızlığının sağlamasıdır.

İkinci kısım 1980'li yıllarda VTYS'lerinde kullanılmaya başlanmış olan ilişkisel sistemlerdir. VT'larında ilişkisel sistemler, bilimsel ve teknik uygulamalar, ses

ve görüntü kullanılan çoklu ortam uygulamaları, ofis uygulamaları, coğrafi bilgi sistemleri uygulamaları gibi alanlarda yetersiz kalmaktadır. Bu yetersizliklerin giderilmesi için 1990'lı yıllarda genişletilmiş ilişkisel, nesneye yönelik, ilişkisel, dağıtık ve çokboyutlu VTYS'leri geliştirmiştir.(Yarımağa, 2000) Bu veritabanı yönetim sistemleri sırasıyla açıklanmaktadır.

### 1.3 İlişkisel Veritabanı Yönetim Sistemleri

İlişkisel veritabanı yönetim sistemi(İVTYS)'lerin gelişmesi, karmaşık içerikli depolama, yeniden düzenleme ve geniş alanda veri işlemleriyle dağıtık fonksiyonlar ve bilgi yönetim sistemleriyle devam etmektedir. İVTYS'leri bilgi sistemleriyle birleşerek işletimsel verinin işlem gücü ve kolaylıkla bütünleşme kabiliyetini sağlayarak sistemin karmaşıklığı uygulama seviyesinde kullanıcıya yansımamaktadır. Aşağıda İVTYS'ne bir örnek olarak öğrenci bilgi sistemi veritabanının ilişkisel gösterimi verilmiştir.

Ogrenci(ogrenci\_no, bolum\_kodu, ogrenci\_ad, ogrenci\_soyad, ogrenci\_sınıf,  
og\_Tckimlik, og\_dogumtarihi, og\_cinsiyeti, og\_tel)

Ogr\_dersnot(ogrenci\_no, ders\_kodu, vize1, vize2, final, ortalama )

Ders(ders\_kodu, ders\_adi, ders\_kredisi, ders\_donemi, bolum\_kodu, ogrt\_kodu )

Ogr\_Gorevlisi(ogrt\_kodu, ogrt\_adsoyad, ogrt\_unvan )

Bolum(bolum\_kodu, fakulte\_kodu ,bolum\_adi)

Fakulte(fakulte\_kodu, fakulte\_adi, fakulte\_dekan )

#### İlişkisel Veritabanı Yönetim Sistemlerinin Yetersizliği

On yıl içerisinde bilgisayar endüstrisinde başarılı değişiklikler olmuştur. VTYS de geleneksel işlem uygulamalarında, İVTYS'lerin yaygın alanlarda kullanıldığı gösterilmektedir. Bununla birlikte İVTYS'leri geleneksel VT uygulamalarından farklı uygulamalarda yetersiz kalmaktadır. Bu uygulamalar:

- Bilgisayar yardımcı tasarım
- Bilgisayar yardımcı yönetim
- Bilgisayar yardımcı yazılım mühendisliği
- Ofis bilgi sistemleri ve çoklu sistemler
- Dijital yayıncılık
- Coğrafik bilgi sistemleri

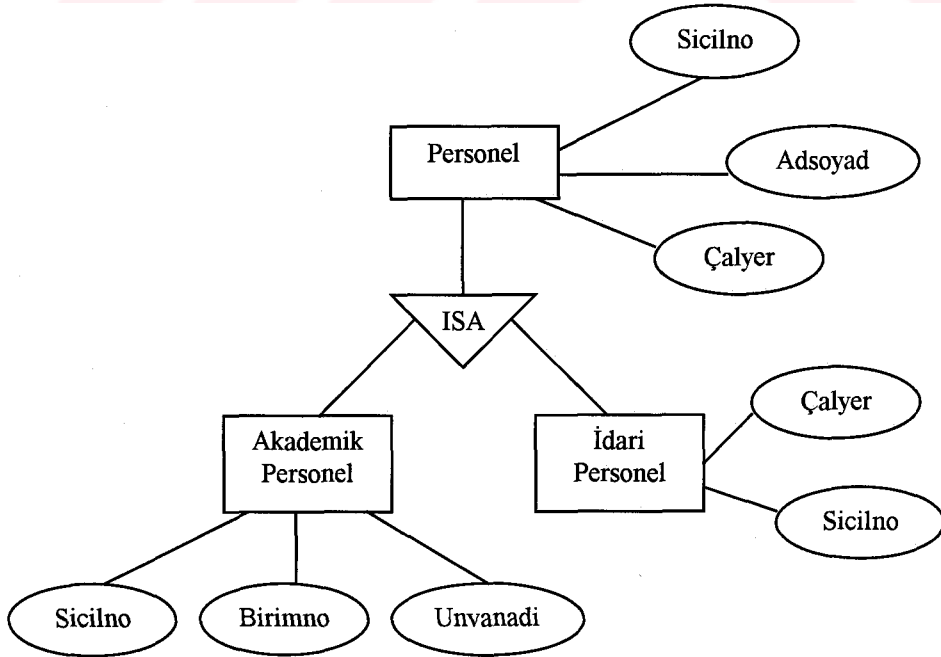
İlişkisel veritabanı yönetim sistemlerinin eksik olduğu konuları belirtmekle birlikte dezavantajlarını da vardır. Bunlar:

- Varlıklarının gösterilmesinde yetersizlik
- Anlamsal aşırı yükleme
- Girişimlerin sınırlandırılması ve bütünlükte zayıf destek verme
- Homojen veri yapısı
- Sınırlı işlemler
- Tekrarlanan sorguların işleme zorluğu
- Özdirenç uyumsuzluğu

#### 1.4 Varlık Bağntı Modelinde Genelleme ve Kümeleme Örneği

##### Genelleme

Varlık kümelerinde ait olma bağıntısı gibi özel bağıntılarda bulunabilir. Alt düzey varlık kümeleri genellenerek bir üst düzey varlık kümesi oluşturulabilir. ISA 'is a kind of' bağıntısı üst düzey bir varlık türü ile bunun alt türleri arasında kurulan özel bir bağıntıdır. ISA bağıntısında üst düzey varlığın nitelikleri kalıtım yoluyla alt düzey varlık türlerine geçer. Bir örnek üzerinde gösterimi Şekil 1.1'deki gibidir.

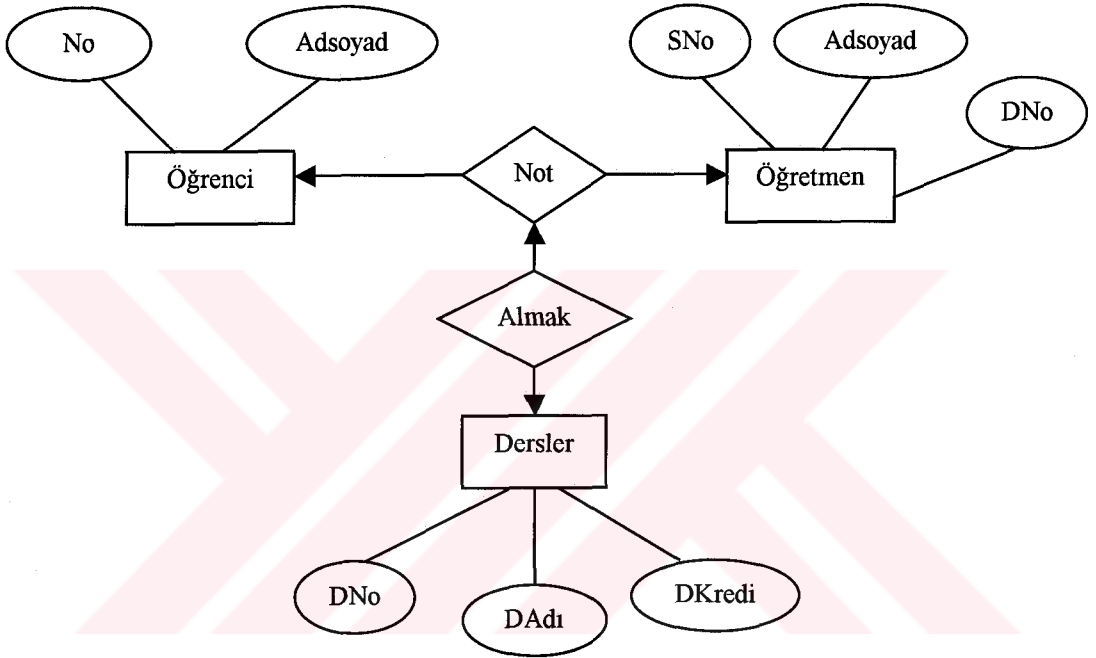


Şekil 1.1 Genelleme örneği



## Kümeleme

Varlık-bağıntı modelinde modelleme kavramlarından biri de kümeleme kavramıdır. Bağıntı kümeleri iki ya da daha çok sayıdaki varlık kümesi arasında kurulur. İki'den çok sayıda varlık kümesi arasında kurulan bağlantılar, çoğunlukla anlamsal açıdan ve işletimsel açıdan uygun çözümler oluşturmazlar. Kümeleme varlık kümeleri ve aralarındaki bağlantının kümelenmesini ve bu kümenin bir varlık kümesi gibi görülmesini sağlar.



Şekil 1.2 Kümeleme örneği

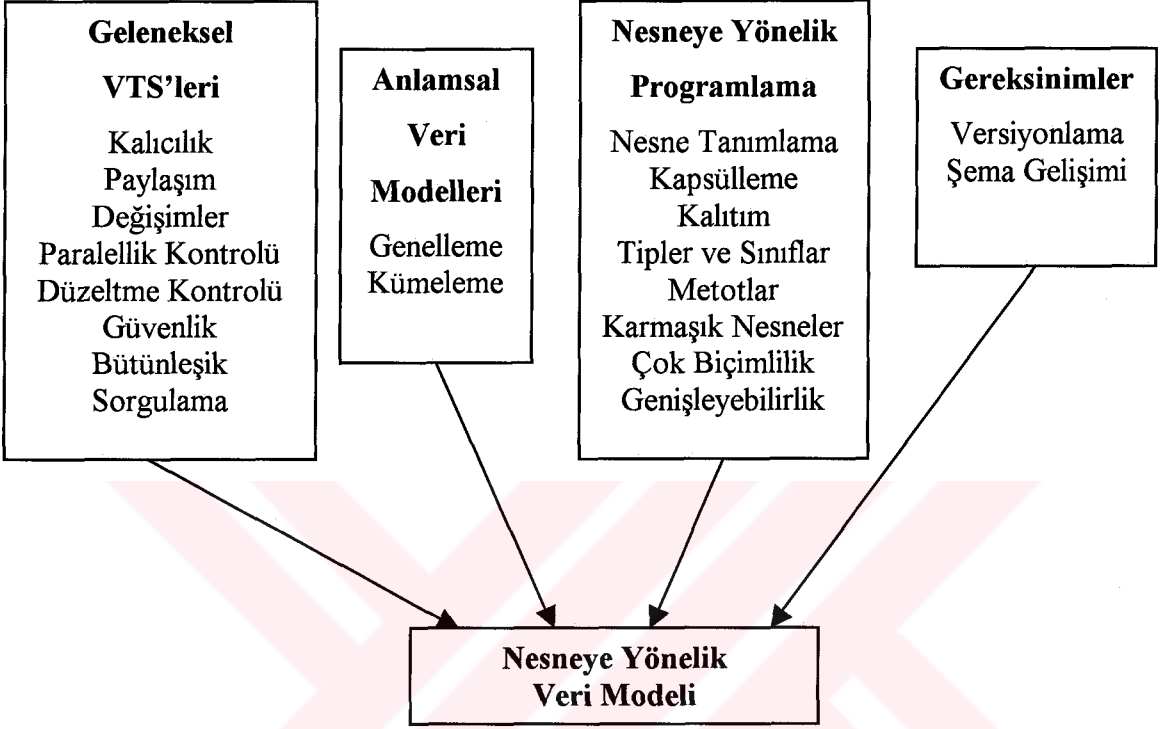
## 1.5 Nesneye Yönelik Veritabanı Yönetim Sistemleri

İlişkisel sistemlerin yukarıda açıklanan yetersizlikleri nedeniyle nesneye yönelik ve nesne ilişkisel veritabanı sistemleri geliştirilmiştir. Nesneye yönelik veritabanındaki birkaç kavramı aşağıdaki gibi açıklarız.

**Nesneye Yönelik Veri Modeli (NYVM) :** Nesneye yönelik programlama dillerinde desteklenen nesnelerin anlamlarını kapsayan bir veri modelidir.

**Nesneye Yönelik Veritabanı (NYVT) :** NYVM tarafından tanımlanmış nesnelerin kalıcı ve paylaşılabilir topluluğudur.

Nesneye Yönelik Veritabanı Yönetim Sistemi (NYVTYS) : NYVT'ı yöneticisidir.



Şekil 1.3 Nesneye Yönelik Veri Modeli esasları (Connolly, 1997)

NYVTS'leri gelişmiş uygulamalarda birçok veri türü için uygun çözümler sağlamaktadır. Bununla birlikte üstünlüklerinin yanında zayıflıklarda bulunmaktadır.

#### **Avantajları:**

- Zengin modelleme kabiliyeti,
- Genişleyebilirlik,
- Empedans uyumsuzluğunun kaldırılması,
- Daha anlamlı sorgu dilleri,
- Şema gelişim desteği,
- Daha uzun süreli işlemler desteği,
- Gelişmiş veritabanı uygulamaları kabiliyeti,
- Performansın artması.

### **Dezavantajları:**

- Genel veri modellerin yetersizliği,
- Deneyimlerin yetersizliği,
- Standartların yetersizliği,
- Sorgu iyileştirmede uzlaşılabilir kapsülleme,
- Nesne düzeyinde kilitlemenin başarımı etkilemesi,
- Karmaşıklık,
- Görüntü desteğinin eksikliği,
- Güvenlik desteğinin eksikliği.

### **Nesneye Yönelik Veritabanı Tasarımındaki İlişki Bütünlüğü:**

İlişkisel nesneye yönelik veri modelinde başvurma özelliği kullanılarak gösterilmektedir. İlişkiler Bire bir (1:1), Bire çok(1:M), Çokla çok(M:N) dır.

1:1 İlişkiler : A ve B nesneleri arasındaki ilişkiyi A nesnesine referans(başvurma) özelliği eklenerek referans bütünlüğü korunarak B nesnesine referans özelliği ekleyerek göstermektedir.

1:M İlişkiler : A ve B nesneleri arasındaki ilişkiyi B nesnesine referans özelliği ekleyerek ve A nesnesine referans kümesini içeren bir nitelik ekleyerek göstermektedir.

M:N İlişkiler : A ve B nesneleri arasındaki ilişkiyi her nesneye referans kümesini içeren bir nitelik ekleyerek göstermektedir.

### **1.6 Dağıtık Veritabanı Yönetim Sistemleri**

Günümüzde bilişim sistemleri ve kapsamındaki VTYS'lerinin çoğunun dağıtık sistemler olması, verilerin birçok yerden güncelleştirilip bir bütünlük halinde kullanımını sağlayan bir sistem oluşturmaktadır. Dağıtık veritabanı(DVT)'ları çoklu VT larına karşı paylaştırılmış verileri kapsar. DVTYS'nin kapsamında dağıtık sorgu işleme(DQP), dağıtık dönüştürme yönetimi(DTM), dağıtık metadata yönetimi(DMM), dağıtık güvenlik yönetimi(DSP) ve dağıtık bütünlük yönetimi(DIM) bileşenlerini kapsamaktadır.

DVTS'de veri, çok kullanıldığı yerlerde depolanır. VT bölüm sayısına bölünerek her bölüm farklı VTYS'lerin kontrolü altında bir veya birden fazla bilgisayarda depolanır. Kullanıcılar uygulamalarla DVT'na erişim sağlar.

**DVTYS özellikleri :**

- Veri bölümlerin sayısında parçalanır.
- Bölümler kopya edilebilir.
- Siteler birbirine iletişim ağlarıyla bağlantılıdır.
- Her sitedeki veri VTYS'nin kontrolü altındadır.
- Her VTYS en az bir genel uygulamaya katılabilir.

**DVTYS modeliyle oluşan sistemlerin başarımını etkileyen etkiler:**

- Sistemin kullandığı ağ yapısının hızlı olması.
- Sistemde kullanılan bilgisayarların çok kullanıcıya eşzamanlı hizmet verebilmesi.
- Kullanıcıların sorgularına kısa zamanda yanıt verilebilmesi.
- Veri sözlüklerinin iyi biçimde güncellenmesi.

**DVTYS Avantajları:**

Dağıtılmış veri ve uygulamalar, merkezleştirilmiş VTYS'leri üzerinde olası avantajlara sahiptir. Uygulama bakımından zor fakat sistem başarımında iyi sonuç veren bir sistemdir. DVTYS'lerini yöneten ve kullanıcılar için sağladığı avantajlar aşağıdaki gibidir.

- Paylaştırılmış ve işlem özerlik birimleri,
- Organizasyonel yapı,
- Paylaşılabilir mantıksal inceleme,
- Gelişmiş geçerlilik,
- Gelişmiş güvenilirlik,
- Gelişmiş performans,
- Ölçeklenebilir ve modüler,
- Düşük maliyette iletişim,
- Veri bağımsızlığının sağlanması.

**DVTYS Dezavantajları:**

- Yönetim ve kontrolde karmaşıklık,
- Maliyet açısından pahalı olması,
- Birden fazla dağıtık sistem kullanıldığı için güvenlik problemleri,
- Bütünleşik kontrolün daha zor sağlanması,
- Standartların sağlanmasındaki eksiklikler,

- Tecrübelerin eksikliği,
- VT tasarımı daha karmaşık ve kompleks olmasıdır.

### 1.7 Veri Tabanı Yönetim Sistemlerinin İş Uygulamalarında Yetersizlikleri ve Veri Yönetiminde Çağdaş Gelişmeler

Geleneksel veritabanlarında kullanılan Çevrimiçi İşlem İşleme(OLTP), işlem tabanlı ve gerçek zamanlı olarak veritabanlarında ekleme, silme, güncelleme gibi işlemlerin gerçekleştiği uygulamadır. OLTP uygulaması maksimum işlem işleme kapasitesi ile tasarlanmasına rağmen özetleme, genelleme gibi işlemler yapılamaktadır. Bununla birlikte OLAP uygulamalarında bu işlemlerin gerçekleştirilmesiyle karar destek için analiz edilmiş bilgi üzerinde anlık sorgulama işlemlerinin tasarlanması elde edilir. OLTP sistemleri ve OLAP sistemleri karşılaştırılmasını çizelge şeklinde vermiştir.

Çizelge 1.1 OLTP sistemleri ve OLAP sistemleri karşılaştırılması

<b>OLTP sistemleri</b>	<b>OLAP sistemleri</b>
Güncel, ayrıntılı veri	Tarihsel, genelleştirilmiş veri
Dinamik veri	Büyük ölçüde statik veri
Tekrarlı işlemler	Ad-hoc, yapısız ve tarihsel işlemler
Tahmin edilebilir örneklerin kullanımı	Önceden bilinmeyen örneklerin kullanımı
İşlemlere yönelik	Analize yönelik
Uygulamaya dayalı	Nesneye dayalı
Günlük kararların desteklenmesi	Şartlara uygun kararların desteklenmesi
Büro, yazı işlemleriyle kullanıcıya hizmet etme	Yönetimsel kullanıcılara hizmet etme

## 1.8 Çokboyutsal Veritabanları

Çokboyutsal veritabanlarıyla ilişkili kavramlar ve iki ana çevrimiçi analitik işleme(OLAP) araçları iki bölümde tanımlanır. Bunlar çokboyutsal OLAP(MOLAP), ilişkisel OLAP(ROLAP) araçlarıdır. OLAP, karmaşık analitik uygulamaları destekleyen bir mimari olarak 1993 yılından Codd tarafından uygulanmıştır. OLAP uygulamaların çoğunluğu uzmanlaşmış çokboyutlu VTYS kullanmada, veri kümelerini sınırlamada ve uygulamaya ait kullanıcı arayüzlerinde uygulanmaktadır. OLAP mimarisi uygulama ve VTYS arasında bir betimleme sağlayan katmanları tanımlar.

OLAP uygulamalarında basit sorgulardan başlayarak karmaşık ve çokboyutsal sorguların gerçekleşmesi için ilişkisel boyutlu veriler ve çokboyutlu veriler kullanılmaktadır.

Yukarıdaki bölümde veritabanlarının gelişim süreçlerini, veritabanı yönetim sistemlerinin bazı avantaj ve dezavantajlarına göre hangi ortamlarda hangi VTYS'ne başvuracağına dair bilgilere yer verdik. Fakat gelişen teknoloji ile birlikte veri üretiminde ve kullanımında artış gözlenmektedir.

Bu nedenle bilgi teknolojisinin gelişim sürecinde veri kaynakları, bu veri kaynaklarının saklanması ve veri kaynaklarına erişim metotlarında yeni teknolojiler ve yeni uygulama alanları oluşmuştur. Bu uygulamalardan veri ambarı ve veri madenciliği uygulaması bir sonraki bölümlerde incelenecektir.

## 2. VERİ AMBARLARI TANIMI, GELİŞİMİ VE TASARIMI

### 2.1 Veri Ambarının Tanımı

Bir önceki bölümde VT'ları ve veri modelleri özelliklerini inceledik. Kullanılan VT'larında daha çok işletimsel veriler bulunmakta ve veriler üzerinde çevrimiçi hareket işlemleri gerçekleştirilmektedir. VT'larında veriler zaman boyutu olmayan, güncel, değişebilen, ayrıntılı verilerdir. Bilgi teknolojisinde bilgi kaynaklarının sayısı son yıllarda çok artmıştır. Bu kaynaklardan sonuçlar çıkararak karar desteğe yönelik bilgiler toplanması çok zor ve güç olup çoğu zaman istenen sonuca ulaşamayan çalışmalar gözlenilmektedir. Özellikle değişik yapıda birbirinden uzak ve çok büyük bilgi kaynaklarında bu güçlük daha da artmaktadır. Bu güçlük ve zorluklarını yenmek için Veri Ambarı(VA)'ları mimarisi oluşturulmaktadır.

Veri ambarı oluşturulmasıyla çok büyük veri yığınlarından özetlenmiş, genelleşmiş verilere ulaşarak onlar üzerinde geleceğe yönelik daha iyi kararlar vermemizi sağlamaktadır. Ayrıca şirketler için günlük rakamların analiz edilmesi ile daha etkili kararlar almakta çok daha iyi sonuç sağlanılır.

VA'larında veriler daha çok değişik kaynaklarda bulunan, değişik görünümlere ve yapılaraya sahip Karar Destek Sistem(KDS)'leri gibi sorgu araçları, Veri Madenciliği(VM) ve Çevrimiçi-Analitik İşleme(OLAP) analizleri kullanılan zaman boyutlu, ayrıntılardan arınmış verilerin gerekli dönüşümlerle tek bir şema altında bütünleştirilmesi ile elde edilir. VT'ları üzerinde ekleme, silme ve güncelleme işlemleri çok sık yapılırken, VA'larındaki veriler üzerinde bu tür işlemler pek yapılmaz. VA'ları kullanım amaçları, içerdikleri verilerin nitelikleri ve veriler üzerinde yapılan işlemler açısından VT'larından farklı olduğundan, VA'ı için kullanılan yapılar, modeller ve teknikler de VT'ları için kullanılanlardan oldukça farklıdır.

### 2.2 Veri Ambarlama Kavramları

Veri ambarı bir çok kaynaklardan elde edilen bütünleşik veri içermektedir. Özet bilgi ile değerlendirilerek uzun zaman diliminde incelenmiştir. Ambarlar diğer çeşit veritabanlarından kapasite olarak daha büyüktür. Anlık veri güncellemesi(Ad-hoc) ve karmaşık sorgular içeren OLTP uygulamalarından farklı olan ambar uygulamaları,

farklı VTYS tasarımı ve uygulama tekniklerinden iyi sonuçlar elde etmede kullanılır. Dağıtık VTYS iyi ölçeklenebilirlik ve yüksek kullanılabilirlik için büyük ambarlara gereksinim duyulmaktadır.

Geleneksel VT'da daha çok işletimsel veriler depolanır ve veriler üzerinde daha çok çevrimiçi hareket işlemleri(OLTP) gerçekleştirilir. VA'daki verilerde KDS, VM ve OLAP uygulamaları kullanılır. VA'ları 90'lı yılların başında, ilk olarak W.H. Inmon tarafından ortaya çıkarılmıştır.

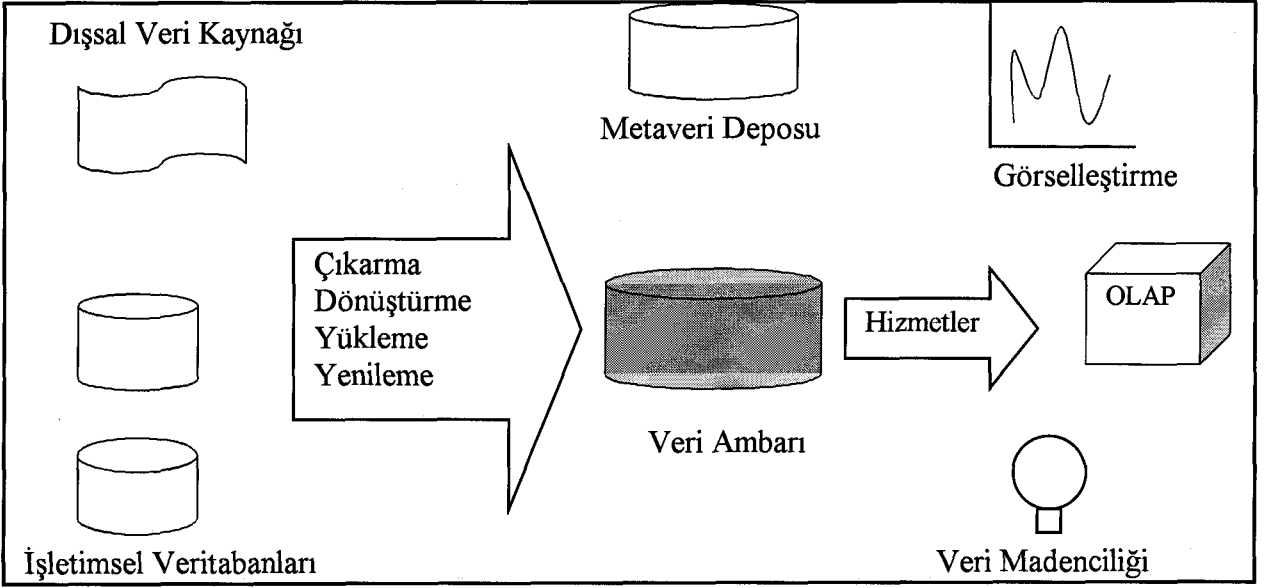
Inmon tarafından yapılan tanıma göre Veri ambarında probleme dayalı olarak uygulamaya yönelik veri yerine karar desteğe yönelik veri depolama gereksinimlerini karşılar. Bütünleşik farklı kaynak sistemlerindeki verinin ortak uygulamaya yönelik olarak birlikte olmasından dolayı bütünleşik veri kaynaklarında kullanıcılara verinin birleşmiş görünümünü gösterir. Zaman Değişkenli ambardaki veri yalnızca doğru zaman aralığında geçerlidir. İstikrarlı gerçek zamanda güncelleşmeyen fakat işletimsel sistemlerde düzenli olarak yenilenebilir. Her zaman veriler VT'na eklenebilir ve bütünlük her zaman sağlanılmaktadır.

### **2.3 Veri Ambarı Oluşturma ve Gelişimi**

Veritabanı şeması farklı kaynaklardan elde edilen verinin bütünleşik topluluğunun elde edilmesi ile tasarlanabilmektedir. Kaynak veri tabanları farklı gruplardan oluşarak, işletimsel veritabanları ve dışsal kaynaklardan veri çıkarımı ilk olarak olasılıklı olarak hatalı veya eşleşmeyen bilginin azaltılmasıyla verinin temizlenmesi sağlanır. Böylece anlamsal hatalar dönüştürülür. Dönüştürülen veri, veri kaynaklarında tablolarla ilişkisel görünümü tamamlanır. Yüklenen veri, ambarda depolanan ve farklı bakış sağlanarak gerçekleşmesi meydana getirilir. İlişkisel VTYS'deki geleneksel görünüşe benzemez. Ambarlar, verinin veritabanında depolanmış görünümüdür. Bu adımları şekil 2.1'de VA oluşum adımları ve genel görünümü gösterilmiştir.

Temizlenen ve dönüştürülen veri, ambara yüklenir. Yeni eklenen işlemler özetlenmiş verinin elde edilmesi ve yerleştirilmesini sağlamaktadır. Çok sayıdaki verinin yüklenmesi düşük hızlı işlem gerektirir. Verinin daha etkili yapılması için veri parçalara ayrılır ve parçalar eşleştirilir. Bu yüzden benzerlik ambar yüklemeye bu önemlidir.





Şekil 2.1 Veri Ambarının genel görünümü

Ambarda veri analizleri ve veri erişiminde farklı araçlar kullanılır. Bunlar OLAP sorgu araçları, veri madenciliği algoritmaları, görselleştirme araçları, istatistiksel paketler ve raporlamalar kullanılır. (Ramakrishnan, 1997)

Veri Ambarı oluşumundaki ve gelişimde kullanılan bileşenlerinin açıklaması aşağıdaki gibidir.

#### İşletimsel Veri Deposu (ODS) :

İşletimsel veri deposu, verinin probleme yönelik, bütünlük, güncel olarak işletimsel kararların desteklenmesi içinde tanımlanmaktadır. ODS / VAM (Veri Ambarı Mimarisi) işletimsel sistem bilgisi, işletimsel sistemlerden elde edilerek ODS içinde birleştirilerek VA'na eklenebilir. VA'na benzer olarak çoklu sistemlerden birleşmiş ve nesne alanlarında gruplanmış veri içermektedir. İşletimsel sistemde küçük tarihsel veri içerir ve işletimsel kullanıcılar tarafından güncellenebilir.

#### Veri Mart Mimarisi :

VA'nın alt kümesi olarak veri mart belirli bölümlerde veya işlem fonksiyonlarının belirli kısımlarındaki gereksinimlerini sağlamayı destekler.

Veri mart, ilişkili VT ve çokboyutlu VT'da özetlenmiş veriyle oluşturulabilir.

Veri Mart ve Veri Ambarının içerdiği özelliklere göre ayrılması:

- Veri Martlar bir bölüm veya iş fonksiyonuyla ilişkili kullanıcının

yalnızca gereksinimleri üzerinde odaklanmıştır.

- Veri Martlar, Veri Ambarından farklı olarak ayrıntılı işletimsel veri içermez.
- Veri Mart, Veri Ambarı ile karşılaştırıldığında onun kadar bilgi içermez. Veri Mart daha anlaşılır ve doğaldır.

### **Bir Veri Mart Yaratmadaki Nedenler:**

Kullanıcının sıklıkla analiz etmeye gereksinim duyduğu verilere erişimi kolaylaştırır. Farklı bölümlerde veya işlem fonksiyonlarında kullanıcı gruplarının bütünlük görünümünü eşleştirilir. Son kullanıcı erişim araçlarının gereksinimleri belirtildiğinde uygun bir şekilde biçimli veri sağlanır. Son kullanıcı erişim araçlarının sayısı, özellikle Veri Mart ve çokboyutsal analiz araçlarının dışsal VT yapılarını gerektirebilir. VA'na göre Veri Mart'ta daha az veri kullanır. Böylece veri temizleme, yükleme, birleştirme işlemleri daha kolaylaşır.

### **Son Kullanıcı Erişim Araçları :**

Veri ambarlarının başlıca amacı önemli konularda karar verme son kullanıcıya bilgi sağlamaktır. Buradaki kullanıcılar son kullanıcı erişim araçlarını kullanarak ambar ile etkileşim halindedirler. Son kullanıcı katılımıyla, özetleme ve süreli raporlarla yüksek başarı gerçekleştirilir.

Buradaki araçları beş grupta kategorize ederiz:

Raporlama ve sorgu araçları

Uygulama gelişim araçları

Yönetici bilgi sistem araçları

OLAP araçları

Veri madenciliği araçları

Son kullanıcı araçlarından OLAP ve veri madenciliği, üniversite bilgi sistemi ambarı oluşturulma uygulamasında gösterimi yapılacaktır.

### **Karar Destek Sistemleri (KDS):**

KDS'leri veri ambarı üzerinde doğrudan analiz yapabilme gereksinimi duymaktadır. Yöneticilerin karar vermesine yardımcı olacak veriye ulaşmasına, özetlemesine ve analiz etmesine yardımcı olmaktadır. Örneğin, bütün sektörlerde firmaların müşterilerini iyi tanıması, hangi ürünlerin kazançlı olduğunun analiz edilmesi, müşteriler ve ürünler arası korelasyon ilişkisinin kurmasında yardımcı olur. Önemli olan bilgiye dayalı bir sistem yaratarak verilere kolay ulaşımı sağlamaktır.

Yönetici Bilgi Sistemi (YBS), bilgiyi düzenleme ve dağıtımını sağlamak için tasarlanmış bir sistemdir. KDS'nde yönetici sistemin içinde görev yapan bir eleman olarak görev yaparken, YBS'de ise sisteminin dışında bir eleman olarak görev yapmaktadır.

YBS yüksek seviyeli karar verme desteğini geliştirmiştir. Buradaki sistemlerin hedefi yönetimin bütün aşamalarında bu desteği sağlamaktadır. Grafikselleştirilmiş karar destek uygulamaları ile verinin görsel görünümünü sağlar.

## 2.4 Veri Ambarı Mimarisi

Yukarıda anlatılan bileşenler ile birlikte düzenlenen VA ağ yapısı çeşitlidir. Merkezleştirilmiş VA, VA ve Veri Mart'lar, Dağıtık VA, Karma VA yapısından oluşmaktadır.

Merkezleştirilmiş Veri Ambarı: VA bütünleşik çoklu işlem birimlerindeki kullanıcılar ile ilgili nesnelere şartlanmış veriyi içerir. Veri Ambarı çapraz işlem bilgi gereksinimlerini destekler. Merkezleştirilmiş VA'nın yapısını oluşturmak kolay olmaktadır. Büyük VA'da ağ yapısı merkezleştirilmiş sistem yönetiminde ve maliyet açısından avantajlıdır.

Veri Ambarı ve Veri Mart'lar: VA, verinin sınırlı alanlarını içeren veri mart ile sıklıkla karşılaştırılır. Veri Mart'lar sıklıkla merkezleştirilmiş ambara bağlanır. İstemciler tarafından VA'dan çekilen veri belirli Veri Mart'lara bağlanır. Kullanıcıya çapraz işlemsel bilgi gereksinimlerini karşılaması için, ambardaki veriye erişim sağlanırken ambar veri istemcisi, istenilen işlemin en iyi şekilde olmasını sağlayan yapısı kullanılır. Bu yapıda Veri Mart'lar sunucunun ikinci katmanında yönetilerek OLAP uygulamaları ve daha gelişmiş işlemsel fonksiyonlar kullanılır.

Dağıtık Veri Ambarı: Bu yapı dağıtık şekildeki verilerden oluşan birden fazla VA'larının bir bütün halinde sistemin devamını sağlayan yapıdır. İşlemler, dağıtık olarak varolan VA'larını desteklemektedir. Eğer veri tek bir yerde ise Fiziksel yapı olarak çoklu VA'ları olmasına karşılık merkezleşmiş VA'dır.(Bontempo ve ark.,1998)

### **Veri Ambarlama Yönetimi ve Gelişimiyle İlgili Karşılaşılan Problemler:**

Veri yüklemedeki kaynakların istenilenin altında tahmin edilmesi: Birçok geliştirici ambardan veri çıkarma, temizleme ve yükleme zamanını ihtiyaç olunan zamanın altında belirtmekten kaynaklanır.

Kaynak sistemlerinde gizlenmiş problemler: Kaynak sistemleri ve VA'nın oluşumunda bazı problemleri gözardı ederek karar verilmesi sonucunda gelecekte problemlerle karşılaşılır.

İstenilen veriye ulaşamaması: Ambar tasarılarında gerekli veriye kaynak sistemlerden ulaşamaz. Çözüm olarak OLTP sistemlerini nitelendirmek veya eksik bilginin elde edilmesini sağlayan bir sistem oluşumuna karar verilmelidir.

Son kullanıcı gereksinimlerinin artması: Son kullanıcı sorgu ve raporlama araçlarını kullanarak isteklerini karşılamaya çalışmaktadır. Bu işlem gerçekleşmediğinde VA'nın nitelikleri ve kapasitesi, kullanıcı taleplerini yerine getirmek için artırılmaktadır.

Verinin bağdaşık hale getirilmesi: Büyük ölçekli VA'da, farklı uygulama alanlarında kullanılan veriler arasındaki farklılıklardan daha çok benzerlikler üzerinde durulmalıdır. Böylelikle veri değeri küçültülür.

Kaynaklarda yüksek talepler: VA donanım olarak diskin büyük bir kısmını kullanmaktadır. Karar destek sistemlerinde kullanılan birçok ilişkisel veritabanlarında çok büyük olgu tabloları yaratılmaktadır. Bu sisteminin boyutunun artmasına sebeptir.

Veri sahipliği: VA'da veri sahipliğinde son kullanıcının tutumunda değişme olabilir. Yalnızca belirli bölüm veya işlem alanında kullanılan ve görünüm sağlayan veriler bulunmaktadır.

Uzun süren projeler: VA işletmede tek bilgi kaynağı göstermektedir VA oluşumu uzun sürelerde oluşmaktadır. Bununla birlikte Veri Mart'lar yalnızca belirli bölümün veya işlevsel alanın gereksinimlerini karşıladığı için daha hızlı oluşmaktadır.

Bütünleşik karmaşıklığı: İşlem süreç zamanının büyük kısmı, değişik VA araçlarının ihtiyaç duyduğu ayrıntılı çözümlerin sağlanılmasında bütünleşik sağlamış olmalıdır.

## **2.5 Veri Ambarlama Teknolojileri**

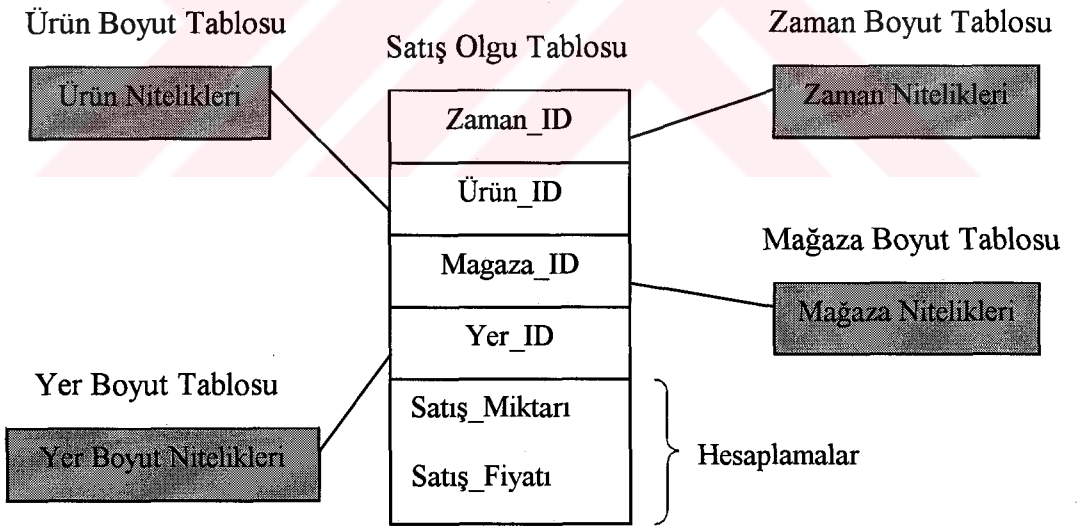
Veri ambarı ve Çevrimiçi Analitik İşlem İşleme(OLAP) karar destek sisteminin temel elemanıdır. VA'nda kullanılan boyutsal modelleme ile ilgili olarak OLAP tasarlanmıştır. Veriye çokboyutsal görünüm sağlamaktadır. İlişkisel OLAP(ROLAP), ilişkisel tablolar arasındaki boyutlar ve olguların oluşmasıdır. ROLAP yapısıyla Yıldız şema gibi veri modelleme teknolojilerinin oluşturulmasıyla verinin ilişkisel görünümü elde edilir. Çokboyutsal OLAP(MOLAP) ise çoklu hiyerarşik yapıya

sahip boyutlar ile veri küpleri elde edilerek her boyutu, veride bir alana karşılık gelen çokboyutlu bir küp olarak bakmayı ve incelemeyi sağlar. Böylece boyut bazında gruplama, boyutlar arasındaki korelasyonları inceleme ve sonuçları grafik, rapor gibi görsellik ön planda olarak sunma olanağı sağlar.(Firestone, 1997)

### Boyutsal Model Şemaları Tasarımı :

#### Yıldız Şema Tasarımı:

Yıldız şema yapısı, bir olgu tablosu ve birçok boyutsal tablolardan oluşur. Boyutsal tablolarının özelliği, bir boyuta ait bütün sıfatlara ait niteleyici bilgilerin tek bir tanım tablosunda bulunmasıdır. Örneğin, zaman boyutu tablosu sadece zamana ait bilgiler tutmaktadır. Olgu tablosu her boyutsal tabloya ait bir yabancı anahtar içermektedir. Olgu tablosunda, ilgili boyuta ait bütün hiyerarşi anahtarların tutulması erişim açısından bizlere kolaylık sağlamasının yanında olumsuz tarafı boyut artırıcı bir etkidir. Yıldız şema yapısı, geleneksel veritabanında kullandığımız ilişkisel veritabanı modellerindeki gibi çokluya-çoklu ilişkileri desteklememektedir. Örneğin, bir bölüm aynı anda, birden fazla birime sahip değildir.(ULEY., 2000)



Şekil 2.2 Yıldız Şema gösterimi

Yıldız şemasının oluşumunu sağlayan boyutlar hiyerarşiktir. Zaman boyut

hiyerarşik olarak yıl, çeyrek, ay, hafta, gün, saat yapısına sahiptir. Böylece saat, gün, hafta, ay, çeyrek, yılın özetlemesidir. Yer boyutunda ise, şehir veya ofis, satış bölgesinin özetlenmesi diyebiliriz.

Yıldız şeması tek bir olgu tablosundan oluşmaktadır. Olgu tablosu ilgili boyutların anahtarları ile ilişkilidir. Yıldız şeması, tasarım kolaylığı sağlamak ve işlem hızını artırması için tasarlanmıştır. Böylelikle şemanın anlaşılması ve şema üzerinden SQL sorgulamalarının yapılması daha kolay olmaktadır.

Veritabanı statik olduğu için anında güncellemeler olmamaktadır. VA'da anlık veri güncellemeleri yapıldığında bu kurallar göz ardı edilebilir.(Dal, 1999) Yıldız şema yapısı şekil 2.2'de gösterimi yapılmıştır.

SQL sorgusu olarak Yıldız şema üzerinde, bir örnek olarak depodaki ürün kategorisi besin olan ürünün 2000 yılının ilk ayında toplam satışı gösteren bir sorgu istenilebilir. Bu sorgunun gerçekleştirilmesi aşağıdaki komutlarla sağlanılmıştır.

```
SELECT Sum(satisolgu.satismiktari) AS SumOfsatismiktari
FROM zamanboyut INNER JOIN (urunboyut INNER JOIN satisolgu ON
urunboyut.urun_Id=satisolgu.urun_Id) ON zamanboyut.zaman_Id = satisolgu.zaman_Id
GROUP BY zamanboyut.yil, urunboyut.kategori, zamanboyut.ay
HAVING (((zamanboyut.yil)="2000") AND ((urunboyut.kategori)="besin") AND
((zamanboyut.ay)="1"));
```

Yıldız şema üzerindeki tablo alandaki ürün kategorisine ait alan bir alt kategoriye ayrılır. Burada birleşme işleminin tersi olarak ayrıntılı veri gösterimi işlemi Drill down SQL sorgusu ile birlikte gösterimi aşağıdaki gibi verilmiştir. 2000 yılının ilk ayında besine ait alt kategori olarak süt ürünleri boyutundaki toplam satış nedir sorgusu işlenmektedir.

```
SELECT Sum(satisolgu.satismiktari) AS SumOfsatismiktari
FROM zamanboyut INNER JOIN (urunboyut INNER JOIN satisolgu ON
urunboyut.urun_Id = satisolgu.urun_Id) ON zamanboyut.zaman_Id=satisolgu.zaman_Id
GROUP BY zamanboyut.yil, urunboyut.kategori, zamanboyut.ay
HAVING (((zamanboyut.yil)="2000") AND ((urunboyut.altkategori)="sütürünleri")
AND ((zamanboyut.ay)="1"));
```

### **Olgu Tablolarının Tasarımı:**

Yıldız şemanın temeli, olgu tablosunun tasarımıyla oluşmaktadır. Olgu tablosundaki özetlenmiş verilere veri ölçüm değeri olarak hesaplama değerleri denilmektedir. Hesaplamalar sayısal değerler olarak hesaplanabilir ve eklenebilmektedir. Olgu tablosunda bütün boyutlar tek birincil anahtara sahiptir. Bu anahtar tabloda tanımlayıcı kolonlardır. Diğer alanlar tam metin tanımlamalarından oluşur.(Uley, 2000)

Yıldız şemanın oluşumundaki önemli kararlardan biri olgu tablosunun atomik derecesidir. Verinin atomik derecesi veya sıklığı, zaman boyutu tarafından tanımlanır. Örneğin, haftalık veya aylık toplam satışın hesaplanması istenilebilir.

### **Olgu Tablosu Tasarımında Etkili Faktörler:**

Her karar destek uygulamalarında zaman periyodu tanımlama gereksinimi.

Ayrıntılı verideki gereksinimlere karşı veri altkümelerinde istatistiksel örneklerin tanımlanması.

Tanımlayıcı kolonlar taşınabilmesi.

Olgu tablosu boyutlarının azaltılabilmesi.

Akıllı ve akıllı olmayan yabancı anahtarların en iyi kullanım tanımlanması.

Olgu tablosunda başlatma zamanına optimal yaklaşımın tanımlanması.

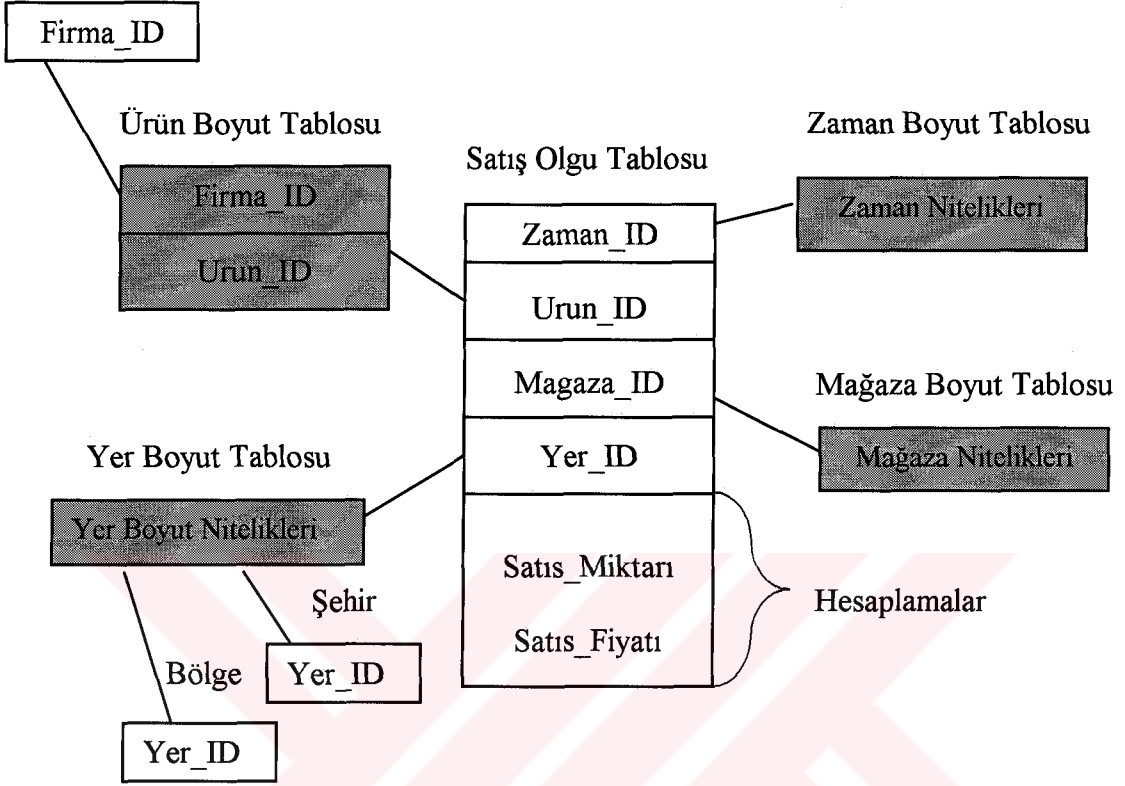
Destek yönetimiyle olgu tablolarının parçalanabilmesi.

### **Boyut Tablolarının Tasarımı:**

Yıldız şemalar tekil boyut tablo içerisinde bilginin denormalize edilmesiyle sorgu performansının hızlandırılmasını sağlamaktadır. Boyut tablolarının hiyerarşik yapısına göre veritabanındaki veriler, yer, zaman, ürün gibi alanlarda sınıflandırma yapısı oluşturulmaktadır.

### **Normalize Edilmiş Yıldız Şema(Snowflake Şeması):**

Normalize edilen yıldız şema, her boyut tablosu kendi içerdiği boyutlardan oluşan bir yıldız şeması çeşitidir. Açıkçası boyutlar, kendilerini tanımlayan boyutlara sahiptir. Normalize edilmiş yıldız şeması boyut tabloları, denormalize edilmiş veri içermemektedir. Bu nedenle bu yapıda birleşmelerin ve düşük sorgulamaların sayısı artmaktadır. Örnek olarak şema şekil 2.3' deki gibi gösterilmektedir.



Şekil 2.3 Normalize Edilmiş Yıldız Şeması

### Yıldızkar Tanesi(Starflake) Şeması:

Yıldız şema(Denormalleştirilmiş) ve snowflake şema(normalleştirilmiş)'lerinin karışımından oluşan hibrit(melez) yapıya sahiptir. Bazı boyutları farklı sorgu gereksinimlerinden oluşan formlarla birlikte gösterilir. Farklı sorgu gereksinimlerini karşılamayı her iki şemadaki formlar sağlamaktadır. Bu yapılar daha karışık olduğu için yıldız şemanın kullanımı daha yaygındır.

### Çokboyutlu ve Çokboyutsal ( MOLAP ) Teknolojisi:

MOLAP teknolojisinde kullanılan veri yapıları, ayrıntılı veriye erişim kolaylığı sağlamakla birlikte çoklu nesne alanlarını destekleme yeteneğinde sınırlıdır. Çokboyutsal veri gösteriminde basit sorgulardan başlayarak karmaşık ve çokboyutsal sorguların gerçekleşmesi için boyutsal veri küpleri kullanılmaktadır. Böylelikle MOLAP araçlarında İVTYS'de olduğu gibi ayrıntılı veriye erişimde kolaylık sağlamaktadır.



İlişkisel veritabanı tabloları, kayıtlar(satırlar) içermektedir. Her kayıt, alanlar(kolonlar) içerir. İlişkisel veritabanında her kayıttta birkaç alan tek tanım satırında oluşmaktadır. Bunun tersi olarak çok boyutlu veritabanı boyutsal dizi içerir. Boyutsal dizide örneğin, ülke, bölge, şehir ve ofis birbiriyle hiyerarşik yapıda dizi oluşturmaktadır.

Veri küpü, veriyi özetlenmiş tablo gibi göstererek boyut özellikler kümesi ve hesaplanabilir özelliklerin kümesi ile bir veya daha fazla toplamlama işlemini gerçekleştirir. Veri küpünde toplam sorguya örnek olarak,

```
SELECT D, Sum(X)
FROM < tablo adı >
GROUP BY D
WHERE Sum()
```

ile tanımlanır. Burada Sum() toplam fonksiyonu, X hesaplanacak olan alan değeri ve D hangi boyut özelliklerin kümesine göre gruplanacağını gösterir. Veri küpleri toplam sorgu işleme hızını artırarak avantaj sağlamaktadır.

Çokboyutsal veri gösteriminde d boyut sayısında 2<sup>d</sup> şeklinde gösteriminin yaratılmasını sağlamaktadır. Örneğin, üç boyut olarak üniversite-akademik personel veri küpünü tanımlayabiliriz.

Üniversite bünyesinde birçok fakülte ve birçok bölüm bulunmaktadır. Uygulama alanı olarak Mühendislik-Mimarlık Fakültesi, Fen Edebiyat Fakültesi, Su Ürünleri ve Ziraat Fakültesi örnek olarak alınmıştır.

Çizelge 2.1 Birimler ve zaman boyutuna ait toplam sayı tablosu

Birimler	Zaman	Toplam Sayı
Üniversite	2002	180
Fen-Edebiyat	2002	156
Mühendislik Mimarlık	2002	41
Su Ürünleri	2002	30
Ziraat	2002	53

Çizelge 2.1’de birimler, zaman ve toplam\_sayı olan üç alana sahip ilişkisel tablo gösterilmektedir. Bununla birlikte verinin iki boyutlu matris şeklinde gösterimi Çizelge 2.2 ‘deki gibi gösterimi daha uygundur.

Çizelge 2.2 Verilerin iki boyutlu matris şeklinde gösterimi

		Birimler				
		Üniversite	Fen-Edb	Müh-Mim	Su Ürünleri	Ziraat
Zaman	Birimler					
	Zaman					
	2000	183	103	27	18	35
	2001	220	126	31	23	40
2002	280	156	41	30	53	

Verilen örnekteki parametrelere göre akademik personelin birime ve yıllara göre toplam sayısının hesaplanmasını veren bir veri küpü oluşturulmuştur.

Akademik Ünvan	Birim			
	Müh.Mim	Arş. Gör.	Yrd. Doç	Prof
Arş. Gör.	14	14	17	23
Yrd. Doç	1	8	8	8
Doç	4	3	2	1
Prof	3	4	3	5
	2000	2001	2002	2003

Şekil 2.4 Verilerin 3 boyutlu şeklinde gösterimi

İlişkisel Veritabanı Yönetim Sistemindeki tablonun çokboyutsal veri halinde gösterimi yukarıda yapılmıştır. OLAP veritabanı sunucuları veriler arasındaki ilişkilerde ve veri depolamada çokboyutsal yapılar kullanmaktadır. Küpün bir kenarı bir boyuta aittir. Bizim örneğimiz birimler, ünvanlar ve yıllardan oluşan bir veri küpünden oluşmaktadır. Çokboyutsal veritabanları görselleştirmeyi ve hesaplama yollarının kolaylıkla anlaşılmasını sağlar.

Çokboyutsal sorguların amacı verilerin özetlenmesiyle ilgilidir. Bütün boyutlar kapsamında toplamlar ve alt toplamların hesaplanmasında etkili çözümler sağlar. Boyutlar hiyerarşik yapıya sahip olabilir. Örneğin, zaman boyutu yıllar, aylar, haftalar ve günler hiyerarşik yapıdan oluşur. Birim boyutu üniversite, fakülte, bölüm, anabilimdalı alanlarına sahip olan hiyerarşik yapıdan oluşur. Boyutlarda önceden tanımlı olan hiyerarşik yapı ön toplamaya izin verir.

Verilen örnekte boyutlar birimler, akademik\_ünvan, zaman'dan oluşmaktadır. Birimler birim\_kodu, akademik ünvan ünvan\_kodu, zaman zaman\_kodu ile tanımlanmıştır. Hesaplama değeri dolu kadro sayısı, birimler üzerinde ortalama sayısı, hangi birimde en çok sayıda personel veya en az sayıda personel bulunan birimlerin hesaplanmasıdır. Bu diziyeye ait şemayı yukarıdaki şekilde gösterilmiştir. Her boyut yapılanmış bir hiyerarşiye sahiptir. Aşağıdan yukarıya doğru bir genelleme vardır.

Çokboyutsal OLAP veritabanı sunucuları ortak analitik işlemleri destekler. Böylelikle çokboyutsal veriler üzerinde analizi sağlayan işlemler yapılabilmektedir.

Roll-up: Aşağıdan yukarıya doğru bir birleşme gerçekleştirir. Örneğin, fakülte-üniversite, ofis- şehir, şehir- ülke arasında roll-up işlemi vardır. Çizelge ile aşamalarının gösterimi verilmiştir.(Gray ve ark., 1997)

Çizelge 2.3 Toplam sayının Roll-up gösterimi

Birim	Yıl	Ünvan	Toplam Sayı by Birim by Yıl by Ünvan	Toplam Sayı by Birim by yıl	Toplam Sayı
Müh-Mim.	2001	Prof.	1	8	
		Yrd.Doç.	7		
	2002	Prof.	0	1	9
		Yrd.Doç.	1		

Drill-down: Birleşmenin tersi olarak ayrıntılı veri gösterme işlemini gerçekleştirir. Örneğin toplam, yıllık, üç aylık, aylık gelirin gösterimi veya bölge, şehir, ilçe, ofise göre ayrıntıya indirgeyerek gösterimi sağlar.

Slice: Kolonlara ve çoklulara göre hesaplamalar gerektirir. Verinin bir bölümü şehirlerin özelliklerine göre toplam gelir üretilebilir.

Dice: Veri kümeleri üzerinde küçük küplere bölme işlemidir. Her şehirdeki ürün çeşitine göre toplam sayının elde edilmesini sağlar.

Veri ambar kullanıcıları, veriye erişimde özetlenmiş veriden sonuçlar çıkarmayı amaçlamaktadır. Özetleme fonksiyonu olarak SQL Server scripting dilinde Transact-SQL işlemlerle SUM, AVG, COUNT, CUBE, ROLLUP işlemler ile özetlenmiş değerlere ulaşılabilir.

ROLLUP ve CUBE, "GROUP BY" ile kullanılır. Bunu bir örnek üzerinde gösterebiliriz.

Ürün\_Depo ( UrunTip: Char, UrunNo: Char, UrunStok: Numeric ) tablosunda

```
SELECT UrunTip AS UrunTip, UrunNo AS UrunNo, SUM(UrunStok) AS Toplam
FROM Urun_Depo
GROUP BY UrunTip, UrunNo
```

Çizelge 2.4 GROUP BY gösterimi

	UrunTip	UrunNo	Toplam
▶	Tip1	UrunA	10000
	Tip2	UrunA	25000
	Tip3	UrunA	34000
	Tip1	UrunB	5000
	Tip2	UrunB	3000
	Tip2	UrunC	10000
*			

```

SELECT  UrunTip AS UrunTip, UrunNo AS UrunNo, SUM(UrunStok) AS Toplam
FROM    Urun_Depo
GROUP BY UrunTip, UrunNo WITH ROLLUP

```

Çizelge 2.5 WITH ROLLUP gösterimi

	UrunTip	UrunNo	Toplam
*	Tip1	UrunA	10000
	Tip1	UrunB	5000
	Tip1	<NULL>	15000
	Tip2	UrunA	25000
	Tip2	UrunB	3000
	Tip2	UrunC	10000
	Tip2	<NULL>	38000
	Tip3	UrunA	34000
	Tip3	<NULL>	34000
	<NULL>	<NULL>	87000
*			

Yukarıdaki SQL komutu ile Tip ve Ürün koduna göre toplam sayısını vermektedir. Eğer Tipe ait ürün kodunda tabloda veri yoksa değeri NULL değeri olarak gösterilir.

```

SELECT UrunTip AS UrunTip, UrunNo AS UrunNo, SUM(UrunStok) AS Toplam
FROM    Urun_Depo
GROUP BY UrunTip, UrunNo WITH CUBE

```

CUBE operatörü ile “süper-özetleme” sonuç kümesi elde edilir. Hiyerarşik alt toplamlar ek olarak çapraz-tablo değerlerini içermektedir. Kolondaki NULL değerleri, satır üzerinde kolonların bütün değerlerin toplamını içermektedir.

Çizelge 2.6 CUBE gösterimi

	UrunTip	UrunNo	Toplam
▶	Tip1	UrunA	10000
	Tip1	UrunB	5000
	Tip1	<NULL>	15000
	Tip2	UrunA	25000
	Tip2	UrunB	3000
	Tip2	UrunC	10000
	Tip2	<NULL>	38000
	Tip3	UrunA	34000
	Tip3	<NULL>	34000
	<NULL>	<NULL>	87000
	<NULL>	UrunA	69000
	<NULL>	UrunB	8000
	<NULL>	UrunC	10000
*			

## 2.6 World Wide Web(WWW)'de Veri Ambarlama

Günümüzde farklı ortamlarda farklı formatta çok büyük değerlerde veriler bulunmaktadır. Bu ortamlarda veri bütünlüğünün sağlanması çok zordur. Bu zorlukların giderilmesi için bazı tekniklerin kullanımı gereklidir. İnternet alanının gelişmesi ile Web üzerinde geniş alanlarda kolay ve hızlı bir biçimde verilere erişim sağlanır. Web üzerinde oluşturulan veri ambarları bütünleşik veritabanları ve veri kaynakları kullanılarak yapılmaktadır. Son kullanıcının analiz erişiminde Web sorgulama teknikleri olarak XML ve Web.sql kullanılmaktadır.

XML'de HTML gibi işaretleme dilleri kullanılmaktadır. Temel fark XML işaretleme etiketlerinin bilginin içeriğini tanımlamak için kullanılır. XML ile herhangi bir uygulama için XML belgesinin içinde bulunacak verinin içeriği ve içerdiği veri tiplerini tanımlayacak uygulamaya özel bir işaretleme dili tanımlanır. Meta veri, veri hakkında veri içerir. XML etiketleri veri hakkındaki meta bilgiyi tanımlamaktadır. Bu teknikler kullanıcılara karar desteğe yönelik son kullanıcının analiz erişimini sağlamaktadır.

Veri ambarlama uygulamaları KDS ve YBS'de kullanılmaktadır. Özellikle şirketlerde son kullanıcının karar verme ve analizlerinde yardımcı olmaktır. Web üzerindeki basit arama motorları webten bilgi elde etmede kullanılır. Veri çıkarımının kolay ve hızlı olmasının yanında bazı dezavantajları da bulunmaktadır. Bazı kısıtlamalar ile veri elde etme seçenekleri azaltılır. Arama motorlarındaki linkler tam olarak doğru sonuç vermeyebilir. Kesin veriye ve sonuca ulaşılmasında dolaylı olarak birbirine bağlı linkler bazen yetersiz gelebilmektedir. Ayrıca bu bağlantı manual olarak gerçekleştirilebilir.

WWW'de sorgu mekanizmasındaki gelişmeler WebSQL, Web3QS, Web Log, Web3QL sorgulama dillerinin gelişimiyle ilgilidir. VT sorgu tekniklerinin birleşmesinde SQL tabanlı olması yatmaktadır. Örnek eşleştirme arama ve kullanıcı eşliğini göstermektedir. Web tabanlı sorgularda çok büyük veriler üzerinde sorgular yapılmaktadır. Böyle bilgi çokluğunda basit anahtar kelime tabanlı arama motorlarında bu işlemi gerçekleştirmek zordur. Web ambarlama daha karmaşık sorgulara cevap verebilmektedir. Web bilgi bağlantı modelini içerir. Web düğümleri ve bağlantıları Web şemaları ve satırları, Web SPJ ( select project join ) işlemleri ve diğer kavramlar VT'larıdaki kavramlara benzemektedir. Web görünümü açısından metinsel cevaplar XML alanları içerisinde geliştirilir.(Varde, 1999)

## 2.7 Mevcut Veritabanları ile Veri Ambarı Tasarımı

Bir veri ambarının tasarımı ve uygulaması için gerekli bilgileri, bileşenleri, mimarisi, teknolojileri hakkında daha önceki bölümlerde bahsedilmiştir. Mevcut veritabanlarındaki analiz servisleri OLAP uygulamaları gelişim araçlarını sağlamaktadır. Analiz servisleri, veri madenciliği teknolojisini kullanarak OLAP küpleri veri analizini gerçekleştirir.

Analiz Servisler araçları ile veri madenciliği işlemleri yerine getirilir. Analizciler, OLAP küplerine karşı Çokboyutsal Anlatım(MDX) dilinde sorgular kullanmaktadırlar.

Boyutsal modellerin tasarımı, OLTP sistemin normalleştirilmiş yapısına gereksinimi vardır. Ve yüksek değerde hızlı işlemleri destekler. İşlemler tek işlem olayını gerektirir. OLTP modelin birçok ilişkili tabloları, web ağına benzemektedir. Tipik boyutsal model yıldız veya snowflake şema tasarımında kullanılır. Veri ambarı boyutsal modelinde raporların gerçekleşimi kolaydır ve etkili bir şekilde özetlenmiş sorgular elde edilir. Veri devamlılığı üzerindeki uygulamalar veri çıkarma, veri dönüştürme ve veri yükleme işlemlerini içermektedir. SQL Server veri değişim servisleri(DTS) birçok değişim tanımlamalarda kullanılan güçlü araçtır. (Riordan, 2001)

OLAP ve veri madenciliği nesnelere yaratma ve geliştirmede, sunucu yönetmedeki uygulamanın gerçekleştirilmesinde karar destek nesnelere kullanılır. Uygulamadaki işlem dizisinde KDS kullanım aşamaları:

- Analiz yöneticisine bağlanması,

- Boyutlar ve küpler içeren bir VT yaratılması,
- Veri kaynağı oluşturulması,
- Boyutlar ve seviyeleri tanımlanması,
- Bir küpün yaratması için boyutlar ve hesaplama değerlerinin belirtilmesi,
- Küp işleminde yapısı ve verilerin eklenmesi.

Küpteki özetlemeler ve verinin depolama işlemlerinin tasarımı veri depolama tipi seçimi ROLAP, MOLAP, HOLAP olarak belirlenerek tasarımı yapılır.

### **3. VERİ MADENCİLİĞİ**

#### **3.1 Veri Madenciliği Tanımı**

Bilgisayar sistemlerinin kullanılmasıyla, çeşitli ortamlarda, çeşitli biçimlerde çok büyük bir veri birikimi oluşmaya başlayıp devam etmektedir. Bu verilerden, kuruluşlar kendileri için çok yararlı bilgileri çıkartabilirler. Fakat bazı bilgiler VT'ları veya VA'ları içinde gizlidir ve doğrudan erişilemez. Faydalı bilgileri elde etmek için geleneksel SQL sorgulama dili veya raporlama araçlarının son yıllarda yetersiz kalması Veri Madenciliği(VM) yöntemlerinin gelişmesine neden olmuştur. Veri madenciliğinin farklı tanımları olarak bilgi keşfetme, örnek keşfetme, bilgi madenciliği, veri tarama gibi tanımlamaları yapılmaktadır.

#### **3.2 Veri Madenciliğinde İzlenen Adımları**

Veri madenciliği işlemi yapılırken izlenilmesi gereken adımlar aşağıdaki gibidir. Başlangıçta gereksinimlerin tespit edilmesi gerekir. Ve sonraki adımlar:

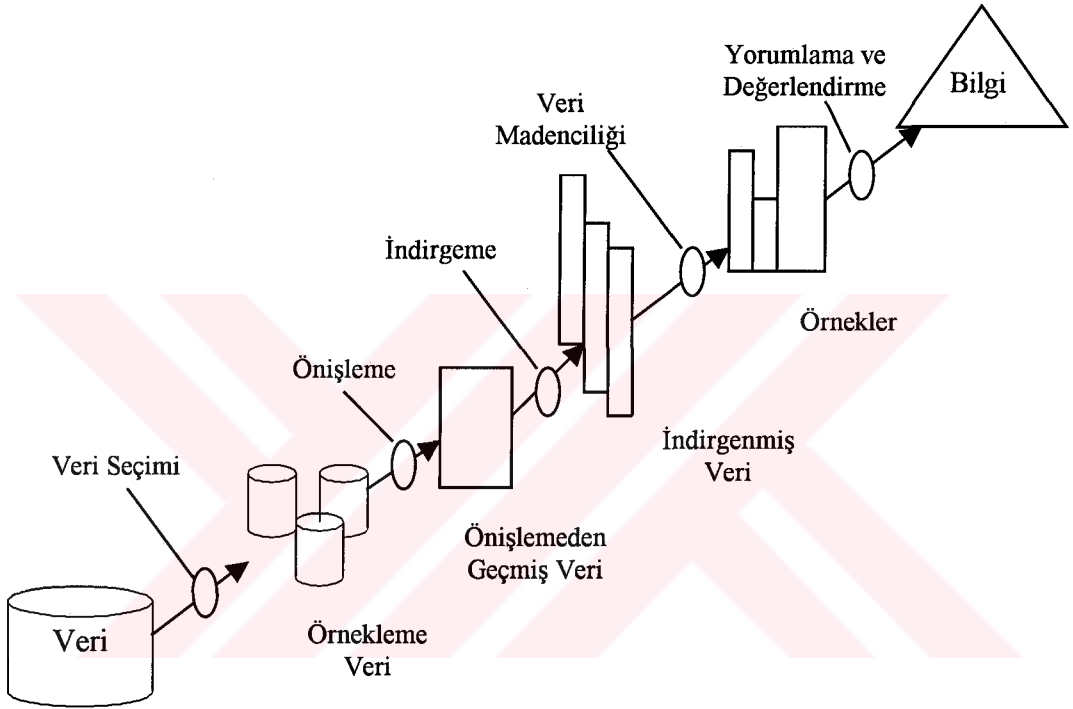
- Problemi tanımlamak,
- Veriyi hazır hale getirmek,
- Veri madencilik modellerinin uygulanması,
- Yararlı sonuçlar elde etmek,
- Sonuçlara göre etkileri tanımlamak,
- Raporların hazırlanması,



- Bir sonraki devrede tanımlanacakları belirtmek,
- Ve istenilen sonucun gerçekleşimi.

Elde edilen sonuçların incelenmesi ile istenilen veri sonuçları elde edilmiş olur.

Çok büyük miktarda veriyi işleme ve veritabanı analizi için veritabanlarında bilgi çıkarımı geliştirilmiştir. Veritabanlarında bilgi çıkarım sürecinde yer alan adımlar gösterilmiştir.



Şekil 3.1 Veritabanı Bilgi Keşfetme Adımları(Fayyad ve ark., 1996)

Veri Madenciliği(VM), veritabanı bilgi keşfetme sürecinde, modelin kurulması ve değerlendirilmesi aşamalarında meydana gelen en önemli kesimi oluşturmaktadır. Örnek vermek gerekirse, üniversite bilgi sistemi veritabanındaki verilerden “bir dersi alan öğrenci sayısı” veya “bir bölümde çalışan personelin isim listesi” gibi bilgiler SQL benzeri araçlarla elde edilebilir.

VTYS bu tür uygulamaların gerçekleştirilmesi için gerekli yapıları ve erişim araçlarını sağlamaktadır. Ancak VTYS'leri ve mevcut araçlar kullanılarak büyük veri yığınlarından “hangi özelliklere sahip öğrencilerin hangi seçmeli dersleri almış olduklarını” ya da “marketlerde hangi ürünle birlikte hangi ürünlerin daha çok satış

yapıldığıının belirlenmesi” gibi bilgileri elde etmesi mümkün değildir. OLAP ile veritabanlarındaki kayıtlara ait sorgular şimdiki durumu göstermektedirler. Fakat VM bizlere bu durumlar sonucunda geleceğe yönelik tahminlerde bulunmamızı sağlar.

VM'nin ilk adımı olarak verinin hazırlanması en çok gayret gerektiren kısımdır. VTYS'nin elde edilmesi veya VA'nın yapımı zordur. Veri üzerindeki işlemler için araçların tanımlanması ve işlemlere başlanması gereklidir. Ve böylelikle faydalı sonuçlar elde edilir.

VM tekniklerinde ürün arama ve ürün gelişiminde değişik seriler ve farklı tablolar arasında birçok değişik kombinasyon ve permutasyon uygulanarak uzun işlemler sonucunda elde edilir. Bilgisayarlarda veri depolanır, veri erişimdeki gelişmeler ile son zamanlarda üretilen teknolojiler kullanıcıya yön gösterir.

VM teknolojisinin esas bileşenleri son yıllarda istatistiksel, yapay zeka ve makine öğrenme gibi arama alanlarında geliştirildi. Aşağıdaki tabloda VM'nin evrimsel gelişim adımları gösterilir. (<http://www3.primushost.com/~kht/text/dmwhite/dmwhite.htm>)

Çizelge 3.1 VM'nin evrimsel gelişim adımları

Evrimsel Adım	İşlem Soruları	Uygulanan Teknolojiler	Özellikleri
Veri Toplama (1960'lar)	Geçen 5 yıldaki toplam birikim nedir?	Bilgisayar, Diskler	Geçmiş ile ilgili değişmeyen veriyi araştırma.
Veri Erişim (1980'ler)	Geçen mart ayında İngiltere 'deki satış nedir?	İVTYS, SQL, ODBC	Geçmiş ile ilgili değişen veriyi araştırma.
Veri Ambarlama ve Karar Destek (1990'lar)	Geçen mart ayında İngiltere'de satış nedir?. Boston'a drill-down biçiminde.	OLAP, ÇBVTYS, VA	Geçmiş ile ilgili aktif bilgiyi araştırma.
Veri Madenciliği (Bugüne kadar devam eder)	Gelecek ay Boston'daki satışlar olası olarak nedir? ve niçin?	Gelişmiş algoritmalar, çokişlemcili bilgisayarlar, Büyük VT'ları.	Geçmiş, aktif bilgi araştırma.

VM bütün problemlere cevap vermeyebilir. Bütün madencilik işlemlerinin elde edilmesi maliyet bakımından yüksektir. Gerçekte madencilik bütün işlemin yalnızca küçük bir adımıdır. Yapılan işlem sırasında bir çok soru ile karşı karşıya kalırız.

Bunlardan bazıları neden madencilik için ihtiyacımız var?. Doğru format içinde doğru veriye sahip miyiz?. İşlemin gerçekleştirilmesi için doğru araçlar mevcut mudur?. Projemizi oluştururken gerekli veri depoları yeterli midir?. Başlangıçta bu sorulara cevap aranmaktadır.

VM'nin ilk adımı olarak özellikle VT'ları ve VT'larında verinin depolanmasıyla başlanılır. Madencilikte kullanılan veriler veri alanlarındaki, ilişkisel VT'ları, çokboyutsal VT'ları ve web üzerinden elde edilen veriler olabilir. VM araçları gelecekteki eğilimleri ve yaklaşımları tahmin ederek bilgi kullanmakta karar vermeyi sağlar. Gizli problemler VT'larında taranarak tahminsel bilgi bulunur. Varolan bilgi kaynaklarının değerini artıran yazılım ve donanım alanında client/server veya paralel işleme bilgisayarların yüksek performansında uygulandığında istenilen veriye ulaşılır.

Veri madenciliğine istatistiksel yöntemler serisi olarak bakmak mümkün olabilir. Örnek olarak nüfus bilgisinin belirli bölgede sonuçları hakkındaki bilgi durumlarını hesaplayabiliriz. Ancak veri madenciliği, geleneksel istatistikten birkaç yönde farklılık gösterir. Veri madenciliğinin amacı, kolaylıkla mantıksal kurallara ya da görsel sunumlara çevrilebilecek nitel modellerin çıkarılmasıdır. Veri madenciliği alanları istatistik, makine bilgisi, veri tabanları ve yüksek performanslı işlem gibi temelleri de içermektedir.

İstatistik metotlardan Lineer regresyon analizinde micro veri analizlerinde doğru sonuçlar, macro veri analizlerinde benzer gösterimle veri küplerinde istatistiksel metotlarla desteklenir.(Shao, 1999)

### **Lineer Regresyon Analizi:**

Analiz üzerinde gerçekleştirilen örnek uygulama olarak üniversitenin herbir fakültesindeki döner sermayede oluşan veritabanları alınmıştır. Veritabanlarındaki tablo alanları, döner sermayenin elde ettiği miktar gelir, harcamalar sonunda geriye kalan miktar birikim bilgileri tablolar halinde belirtilerek Çizelge 3.2'de gösterilmektedir. Elde edilen verilerin ortalama gelir ve ortalama birikimleri fakülteye göre sınıflandırılarak Çizelge 3.3'de gösterilir. Lineer regresyon analizi Çizelge 3.2'de micro veriler ve Çizelge 3.3'deki macro veriler kullanılarak gelirler üzerinde birikim lineer regresyon analizi yapılır. x bağımsız değişkenine karşılık gelen y bağımlı değişkenler arasında doğrusal, karesel, üstel bağıntılar bulunarak doğru denklemi oluşturulur.

Çizelge 3.2 Gelir-birikim verileri

DSermaye	Gelir	Birikim	DSermaye	Gelir	Birikim
CA	21,000	1,400	TN	14,000	14,000
CA	28,000	3,000	TN	19,000	19,000
CA	32,500	3,150	TN	31,000	31,000
CA	41,000	4,200	WY	21,000	21,000
CA	54,000	6,500	WY	32,000	32,000

Çizelge 3.3 Gelir-birikim verilerin ortalama tablosu

DSermaye	Avg(Gelir)	Avg(Birikim)
CA	35,300	3,650
TN	21,333	2,133
WY	26,500	4,450

Çizelge 3.2'deki gelir ve birikim verileri arasındaki doğrusal bağıntı;

x: gelir değerleri

y: birikim değerleri

n: toplam değerlerin sayısı değişkenlerini tablodaki veriler elde ederiz. Lineer Regresyon denkleminin bulunması için aşağıdaki formüller adım adım gerçekleştirimi yapılır.

$$S_{xy} = \sum xy - \frac{(\sum x)(\sum y)}{n} \quad xy'e \text{ göre standart sapma}$$

$$S_{xx} = \sum x^2 - \frac{(\sum x)^2}{n} \quad \text{xx'e göre standart sapma}$$

$$S_{yy} = \sum y^2 - \frac{(\sum y)^2}{n} \quad \text{yy'e göre standart sapma}$$

$$a_1 = \frac{S_{xy}}{S_{xx}}$$

$$a_0 = \bar{y} - a_1 \bar{x}$$

formüllerden elde edilen sonuçlar:

$$S_{xy} = 163.982.500$$

$$S_{xx} = 1.247.025.000$$

$$S_{yy} = 31.322.250$$

$$a_1 = 0.1314$$

$$a_0 = -504.495$$

$y = a_0 + a_1 \bar{x}$  denkleminde elde edilen sayı değerleri yerine yazıldığında

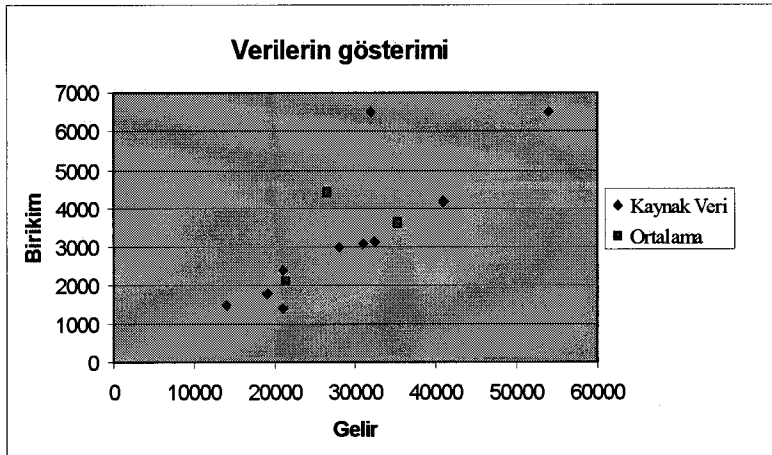
Çizelge 3.2' deki micro veriler kullanılarak regresyon denklemi

$$y = -504,5 + 0,13\bar{x} \text{ elde edilir.}$$

Aynı işlemler Tablo 3.2'deki macro veriler için uygulandığında

$$y = 992.7 + 0.009\bar{x} \text{ regresyon denklemi elde edilir.}$$

Elde ettiğimiz lineer regresyon denkleminin Çizelge 3.2 ve Çizelge 3.3'deki veriler kullanılarak grafiksel gösterimi yapılır.



Şekil 3.2 Regresyon Denkleminin Grafik Gösterimi

Kaynak veri ve ortalamaya göre elde edilen regresyon denkleminin grafiksel gösterimi verilmiştir. Elde edilen formül sonucunda kaynak veriler ve ortalama değerlerinin karşılaştırılması gösterilir. Ve başka X değerine göre Y değerinin elde edilmesini sağlar.

Veri Madenciliği olarak Eşleştirme Kuralları, Sınıflandırma, Regresyon, Zaman İçinde Sıralı Örüntüler, Benzer Zaman Sırası ve Fark Saptanması uygulama alanları örnek olarak verilebilir. Bu uygulama alanları yerine getirilirken bazı teknikler, işlemler kullanılmaktadır.

### 3.3 Veri Madenciliği İşlemleri

VM süreci sonunda elde edilen bilgiler kurallar biçiminde ifade edilerek istenilen bilgi çıkarımı gerçekleşir. Elde edilen kurallar iki değişkenin eşleştirme derecesini gösterir, ya da veri önceden tanımlanmış sınıflara paylaştırılır veya veri tanımlayan sonlu sayıda kümeye ayrılır. Bu kurallar veri üzerinde belirli bir tekniğin uygulanması sonucunda elde edilir. Elde edilen bilginin doğruluğu veri analizi için kullanılan yöntemlere büyük ölçüde bağlıdır.

VM yöntemleri iki sınıfta toplanabilir. Keşfe dayalı ve doğrulamaya dayalı olmak üzere incelenebilir.

Keşfe dayalı yöntemler gizli örneklerin çıkarılması işlemidir. Örneğin, banka çalışanlarının yaş ortalamaları üzerinde yapılan madencilik sonucunda banka çalışanlarının oluşturduğu sistemde ortalama yaş grubu 47 olarak söyleyebiliriz. Keşfe dayalı yöntem başka bir örnek kurallar üzerine uygulanarak verilebilir. If Meslek=Sporcu Then Yaş<30, 71% güven aralığı ile bu koşul sağlanır. Bunun anlamı 100 sporcudan 71'inin yaşı 30'dan küçüktür. Eldeki verilerden örneklerin indirgenmesiyle kullanıcılar hakkında bilgiler elde edilir. Keşfe dayalı algoritmaları kendi arasında şöyle sıralanabilir: Sınıflama- hiyerarşik, Kümeleme-gruplama, Ardışık Zamanlı Örüntüler, Birliktelik Kuralları, Regresyon Analizi.

Doğrulamaya dayalı yöntemlerde veritabanından önceden elde edilen örneklerden sonuçlar elde ederek bilinmeyen değerler tahmin edilebilmektedir. Veritabanından örnekler çıkarımı sonucunda geleceğe yönelik tahminlerin elde edilmesinde kullanılır. Tahmin edilen model böylece kullanıcıya sunulur. Ayrıca

çıkarılan veriden örnekler bulurken, tahmin edilen model yeni veri kümelerinde tahminleme değerlerine örnekler sunar. Doğrulamaya dayalı yöntemde bilgi çıkarımından önce eldeki verilerden yola çıkarak yeni hipotez geliştirilir ve geliştirilen hipotezde sistemdeki veriler kullanılarak hipotez doğrulanmaya çalışılır. Ve böylelikle istenilen sonuca ulaşılır. Bu sisteme yine yaşlara göre bir örnek verilirse, sporcuların yaşı 30'dan küçüktür diyebiliriz. Bize bu hipotez sonucunda gelen değerlerde eğer kişi sporcu ise onun yaşını kolaylıkla tahmin edebiliriz.

VM yöntemlerinin dışında kullanılan bazı teknikler olarak makine öğrenme, istatistik, sinir ağları ve genetik algoritmalar, görselleştirme, yüksek performanslı paralel işlemlerle ilgili gelişmeler örnek verilebilir. VM otomatik bilgi çıkarma yaklaşımı yapay anlayış ve makine öğrenme alanlarındaki çalışmalardan etkilenmiştir

### **Veri Madenciliği Faaliyetleri ve Kullanıcıları:**

Veri Madenciliği faaliyetleri, farklı üç kullanıcı sınıfı tanımlar. Bunlar yöneticiler, kullanıcılar ve analizcilerdir.

Yöneticiler, en üst seviyedeki görünüme sahip ve diğer gruplardan daha az bilgisayar başında vakit geçiren kısımdır. İşlemlerin nasıl veya ne şekilde gerçekleşmesiyle ilgilenmeyen gerekli bilgiyi görsellik açısından görebilmek istemektedir. Son kullanıcılar çapraz-tablolama işlem sonuçlarını kullanırlar. Çoğunlukla son kullanıcılar ile yöneticilerin kurallarının aynı olduğunu düşünebiliriz. Analizciler elde edilen verilerinin kullanımıyla ilgilenirler. Onlar genellikle finans analistleri, istatistikçiler, danışmanlar veya veritabanı tasarımcıları olarak sonuçları düzenleyenlerdir.

### **3.4 Veri Madenciliği Teknolojileri**

Veri madenciliği teknolojileri, veritabanı yöneticisi, veri ambarlama, istatistik, makine öğrenme, karar ağaçları, görselleştirme ve paralel hesaplama olarak ifade edilebilir. Buradaki teknolojilerden makine öğrenme algoritmaları ve istatistiksel paketler son yıllarda var olarak veri yönetimi ve verinin organize edilmesinde önemli rol oynamaktadırlar.

Makina öğreniminde VM algoritmalarında kullanılan örneklem boyutu, makina öğreniminde kullanılan veri boyutuna nazaran çok büyüktür. Karşılaştırmak yapılırsa

makina öğreniminde kullanılan örneklem boyutu 100 ile 1000 arasında değişirken VM algoritmaları milyonlarca gerçek hayat nesnelere üzerinde uğraşmaktadır ki bunların karakteristiği boş (null), artık, eksik, gürültülü değerler olarak belirlenebilir. Makine Öğrenme(MÖ), veriden öğrenme rolleridir. MÖ önceki tecrübeler, deneyimler hakkında bazı performans kriterlerine rağmen bilginin öğrenilmesidir. Örneğin, satranç oyunu oynamak daha önceki hamleler değerlendirilerek onlar üzerinde öğrenme gerçekleşir. Daha çok sembolik ve sınıflandırılmış veri türlerini içermektedir.(Thuraisingham.,1999)

Görselleştirme, VT'larındaki verinin grafiksel gösterimi şeklindedir. Veritabanlarındaki veri sayısal satırlardan oluşmaktadır. Görselleştirme araçları, elde edilen veriyi grafik formu içerisinde göstermektedir. Görselleştirme modeli bir boyutlu gösterimle birlikte çok boyutlu gösterim değerlerini de saklamaktadır.

### 3.5 Veri Madenciliği Yöntemleri

Veri Madenciliği yöntemlerinin kullanılması VT'larında ve VA'larında büyük değerlerde depolanan verinin analizinde ve madenciliğinde, etkili araçların gelişiminde önemli rol oynar. Veri madenciliğine ait yöntemleri kısaca aşağıdaki gibi açıklayabiliriz.

#### 3.5.1 Sınıflama

Sınıflama, verilerin istenilen alt sınıflar içinde gruplaşması işlemidir. Sınıflama, sınıf değerlerinin başlangıç değerlere sahip VT'da her yeni eklenen kaydın daha önceden tanımlı sınıfa yerleşmesinde kullanılır. Sınıflamada veriler hiyerarşik yapıya sahiptirler. Sınıflamaya ait kuralları aşağıdaki gibi gösterebiliriz.

$$(l_1 \leq X_1 < h_1) \wedge \dots \wedge (l_k \leq X_k < h_k) \Rightarrow Y = c$$

$X_1 \dots X_k$  nitelikleri tahmin etmede veya sınıflamada kullanılır.  $Y$  nitelik değeridir. Sınıflama kuralının her iki taraf bazı ilişki satırlarında niteliklerin koşulları gibi düşünülebilir.

Sınıflama kuralının sağ tarafı  $Y = c$ 'dir.  $c$  olasılık değerlerin kümesidir. Her sınıflama kuralında  $X_1 \dots X_k$  satırlarında  $Y$  değeri tahmin edilir. Sınıflama kuralını daha genel bir fonksiyon olarak yazılması istenirse,



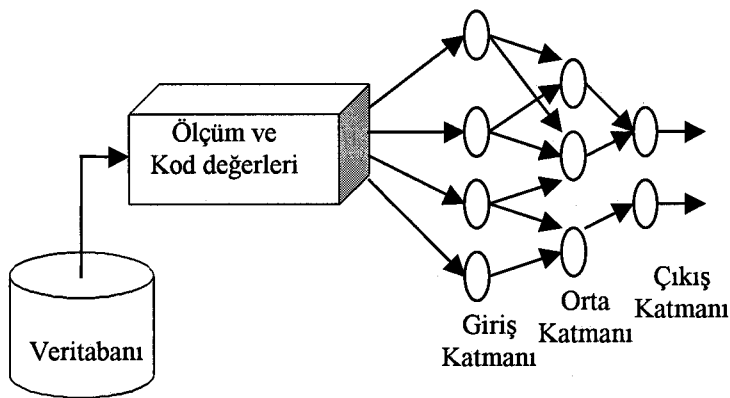
$$(l_1 \leq X_1 < h_1) \wedge \dots \wedge (l_k \leq X_k < h_k) \Rightarrow Y = f(X_1, \dots, X_k)$$

şeklinde yazılabilir.  $f$  bir fonksiyondur. Böyle kurallar regresyon kuralı olarak isimlendirilir. Örneğin, Fakülteleere ait bölümler tablosu bulunmaktadır.  $n$  tane kayıta sahiptir. Bu kayıtlar Fakülte\_Id'lerine göre sınıflandırılır. Mühendislik-Mimarlık fakültesi altında yeni bir bölüm açıldığından birincil anahtar(Id) numarasına göre ekleme işlemi gerçekleşir. Bu örneğe Fen Bilimleri ve Sosyal Bilimler Enstitülerini örnek olarak verebiliriz. Birkaç sınıflama modelleri :

**Karar Ağaçları:** Regresyon analizi konusunda gösterilmiştir.

**Yapay Sinir Ağları:** İnsan beyinin örnek-bulma kapasitesini taklit etmektedir. Sinir ağları, önceki gözlemlerden yeni gözlem tahmin etmede beyin nöronları fonksiyonu ve kavramsal sistemde öğrenme işlemlerinden sonra modellenen analitik tekniklerdir. Katmanlar, birbiriyle bağlı nöronlardan oluşmaktadır. Bir katmandaki  $j$ . birime gelen toplam giriş, önceki katmandaki birimlerin  $y_i$  çıkışlarının bağıntılar üzerinde  $W_{ij}$  ağırlıkları ile hesaplanmış ağırlıklı toplamıdır.

Her nöronun temel fonksiyonu, giriş değerlerini değerlendirme, giriş değerlerinin birleşmesinde toplamın hesaplanması, eşik değeriyle toplamın karşılaştırılması ve çıkışların tanımlanmasıdır. Şekil 3.3'te giriş nöronların katmanlara bağlantıları gösterilmektedir.



Şekil 3.3 Yapay Sinir Ağları Modeli

Nöron modeli, ilk olarak daha önce eğitilen ağ üzerinde tahminler yapılır. Eğitime süresince bazen öğrenme işlemlerinin kontrolünde öğrenme aşaması süresinde görüntüleme veri kümesi kullanılır.

Her nöron ağırlık kümesi giriş sinyal güçlerinin birleşmesidir. Girişler pozitif veya negatif olarak nörona katılır. Öğrenme, ağırlıklar değişimleriyle meydana gelir. Girişler genellikle ölçeklenmiş ve normalleştirilmiştir.

Öğrenme aşamasında ağ kümeleri ara katmanın davranışını tanımlayan ağırlıklardır. En çok kullanılan backpropagation yaklaşımı, ağırlıklar, ağın tahminleri üzerine ayarlanır. Yanlış tahminler bağlantılardaki eşik değerlerini indirger. Sinir ağları, veri kümeleri üzerinde fonksiyonların sonuçlarına göre tahmin edilebilir olması nedeniyle öğrenilebilir.

K-En yakın Komşu: En iyi bilinen veri bulma yaklaşım örneği “en yakın komşu bulma” metottudur. Yeni veriler veri kümesine eklenerek yeni kayıt tahmininde kullanılır. Örneğin, banka servisindeki tahmin edilen müşterinin yaş ve geliri ile var olan müşterilerin yaş ve geliri karşılaştırılır. En yakın olan tahmin seçilir.

Sınıflama-tabanlı öğrenmede uygulanan algoritmalar:

ID3 Algoritması(Quinlan86): Karar ağacı oluşturmadaki algoritmadır. Özellikleri deneme değerleriyle nesnelere sınıflanmasının tanımını sağlar. Özelliklerin açıklamasından başlayarak nesne kümeleriyle devam eder. Böylelikle yukarıdan aşağıya doğru dallanan ağaç yapısı oluşur. Ağacın her yaprak düğümünde bir özellik test edilir ve sonuçlarda nesnelere ayrıştırılarak nesne kümesi oluşturulur.

C4.5 Algoritması : Bu algoritma Quinlan tarafından önerilmiştir. Bu algoritma verinin tekrarlanan bölümleriyle verilen veri kümesindeki sınıflanma-karar ağacını üretir.

SLIQ (Araştırmalarda Tahminsel öğrenme) Algoritması, Naive-Bayes, Logistik Regresyon, Rule induction de bunlardan bazılarıdır.

### 3.5.2 Kümeleme

Kümeleme analizinde veritabanı, altkümelere ayrılır. Her bir kümede yer alan elemanlar dahil oldukları grubu diğer gruplardan ayıran ortak özelliklere sahiptir. Kümeleme, farklı sınıflama algoritmalarını kapsar. Verinin nasıl düzenleneceğine ait genel sorular ve sorgu alanı içerir. Kümeleme tekniklerinin amacı küme üyelerinin

birbirine çok benzediği, ancak özellikleri birbirinden çok farklı olan kümelerin bulunması ve veri tabanındaki kayıtların bu farklı kayıtlara bölünmesidir. Örneğin, biyologlar hayvanların farklı türlerini inceleyen kişilerdir. Bunu yaparken hayvanlar arasındaki varolan farkları tanımlar. Üniversite Bilgi Sistemine bir örnek verilirse, Fakülte Bilgi Sistemine göre bir sorgu listesi oluşturmak istenirse, bölümlerin çalışma alanları ve ortak çalışma alanları belirlenerek bu çalışma alanlarında uygulamaları olan akademisyenlerin belirlenmesi kümeleme analizine bir örnek olarak verilebilir. Kümeleme geniş alanda bir istatistik çalışmadır. Geniş ve farklı veri dağılımlarında kümeler bulunur.

Kümeleme analizine ait algoritmalar: K-Yakın komşu bulma, Kohonen Ağları, İki adımlı kümeleme yöntemleri örnek olarak verilebilir.

### 3.5.3 Birliktelik Kuralları ve Ardışık Zamanlı Örüntüler

Birliktelik kurallarının satışa ait bir örnek üzerinde tanımı yapılırsa bir alışveriş sırasında veya birbirini izleyen alışverişlerde müşterinin hangi mal veya hizmetleri satın almaya eğilimli olduğu belirlenebilir. Bunun sonucunda daha fazla ürünün satılmasını sağlama yollarından biri bulunmuş olmaktadır.

Ardışık zamanlı örüntüler ise birbirleriyle ilişkisi olan ancak birbirini izleyen dönemlerde gerçekleşen ilişkilerin tanımlanmasında kullanılır. Birbiriyle ilişkili olan ürünlerin birlikte alınması veya satılmasının yüzdesine göre hesaplanan değerler destek ve uyum değerleridir. Burada destek değerinin anlamı, veride belirlenen bağının ne kadar sık olduğunu belirtir. Uyum değeri ise bir değerle birlikte hangi olasılıkla diğer değere sahip olunacağını belirlemektedir. Bu tanımı öncel=>sonuç şeklinde belirtebiliriz. Bu koşul bize ürünle birlikte alınan diğer ürünün alınmasını göstermektedir. İlk alınan ürün öncel bunun sonucunda alınan diğer ürün ise bunun sonucunda oluştuğu için bu değere sonuç değeri denilir.

Bu yöntemin amacı hem öncel hem de sonuç örnekleme yüzde kaç tarafından sağlandığını belirtir. Uyum kuralı ise, öncel kesimin sağlandığı durumların örnekleme yüzde kaçında kuralın diğer sonuç kesiminde sağlandığının bulunmasını sağlar. Destek ve uyum değerlerini bulmamızı sağlayan denklem aşağıdaki gibidir.

X ve Y bir ürüne ait değerlerdir.

$P(X \text{ ve } Y) = X \text{ ve } Y \text{ mallarını satın almış müşteri sayısı}$

$P(Y) = Y$  mallarını satın almış müşteri sayısı

Destek:  $P(X \text{ ve } Y) = P(X \text{ ve } Y) / \text{Toplam müşteri sayısı}$

Uyum:  $P(X \setminus Y) = P(X \text{ ve } Y) / P(Y)$

Bunu öğrenciler üzerindeki bir örneklem kümesinde uyguladığımızda, matematik dersini alan öğrenciler beden eğitimi ve spor dersini almıyor.

$P(X \text{ ve } Y) = \text{Matematik ve spor dersini alan öğrenci sayısı} = 20$

$P(X) = \text{Matematik dersini alan öğrenci sayısı} = 25$

Toplam öğrenci sayısı = 40

Destek =  $20/40 = 50\%$

Uyum =  $25/40 = 80\%$  sonucunu elde ederiz.

Eşleştirme sorgusunu yukarıda bahsettiği biçimde ele alan bir kaç algoritma vardır. Bu algoritmalarından en bilineni Apriori algoritmasıdır.

Apriori Algoritması :

Veritabanında sıklıkla geçen veri kümelerini bulmamızı sağlayan algoritmadır.

Bu algoritmanın adımları:

1) Veri öğelerinin sahip olduğu birleşmeler sağlanılarak minimum desteğin yukarısında işleme sahip olunur. Minimum desteği sağlayan kümeler o işlemden üretilen sık geçen veri kümesi oluşturulur.

2) Bu işlem yeni bir sık geçen veri kümesi bulunmayana kadar devam eder.

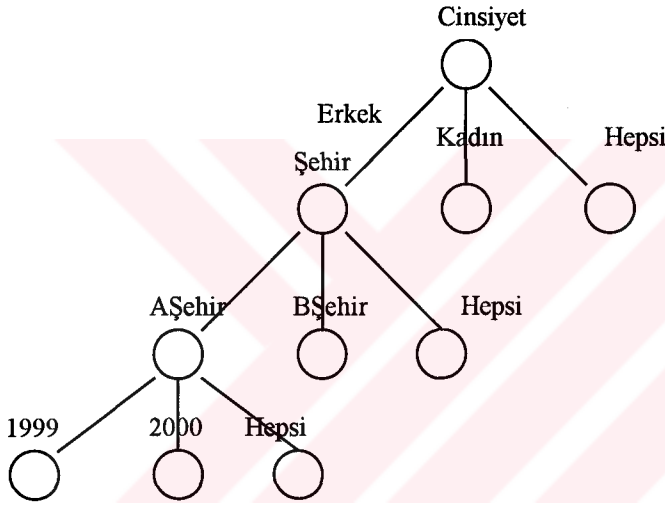
Birliktelik kuralları ile sınıflama kuralları arasında farklar bulunmaktadır. Birliktelik kuralları, elemanları arasında bir mantık döngüsü vardır. Süreklilik değerine sahiptir. Örneğin, A, B, C olaylarımızda eğer A olursa B veya(ve) C olabilir. Sınıflama kurallarında kesiklilik ve bağımsızlık vardır. Elemanlardan biri olmasa bile sınıf şekli dağılmaz. Elemanların birbirine göre hiçbir döngüsel olayı yoktur. Örneğin, X bölümünde okuyan öğrenciler arasında bir süreklilik bağıntısı yoktur. Bu bölümden bir öğrenci kayıtlı silinse, bu sınıf etkilenmez.

### 3.5.4 Karar Analizi

VM karar analizi karar kurallarını ve karar ağaçlarını desteklemektedir. Karar kuralını bir önceki bölümde nüfus sayımı örneği üzerinde vermiştik. Aynı veriler üzerinde karar ağacı oluşturmayı inceleyebiliriz. Nüfus sayımına ait veritabanında karar kurallarının oluşumu gelirlerin hesaplanması, yıl, durum ve cinsiyet özelliklerine ait

regresyon denklemi oluşturulmuştur. Burada IF-THEN kurallarını yazılabilir. Şekil 3.4'te karar ağacı yapısı verilmiştir. Ve karar ağacı şu kurallara karşılık gelmektedir. IF (( 1999<yıl<2002) and (şehir="Çanakkale") and (cins="erkek")) THEN (Birikim=a+b\*Gelir) olarak verilir. Bu şekilde kural çıkarımı, verilerin indirgenmesiyle istenilen sonucunun geçerliliği sağlanır. Ağaçta ilerleme yönü kökten yaprağa doğru kurallar gerçekleşir.

Karar ağacında, her yaprak olmayan düğüm boyutların niteliklerini ve her kenar niteliğin alanlarındaki farklı değerleri gösterir. Her yaprak düğüm regresyon denklemiyle ilişkilidir. Yaprak düğümle kök düğümden bir yol regresyon kuralı belirler. Şekil 3.3' de regresyon ağaç örneği şekillenir.



$$S = -3222 + 0,42 * 5$$

Şekil 3.4 Karar(Regresyon) Ağacı

Veri madenciliği karar kuralları veya karar ağaçları işlemleri üç alana bölünür: özetlenmiş veri elde etme, boşluk indirgeme ve tahminlemedir.

### 3.6 Öğrenmedeki Modellerin Kurulması

Bir problemin çözümünde işlemler, kavramsal sistemlerle uyumun sağlanmasıyla sistem öğrenmeli olarak tanımlanır. Öğrenme, uygun sınıflar oluşturulmasıyla model değişim fonksiyonların sınıflar üzerinde işlenmesi olayıdır. Burada değişik öğrenme yöntemleri uygulanarak model kurulumu

denetimli(*Supervised*) ve denetimsiz(*Unsupervised*) öğrenimin kullanıldığı modellere göre farklılık göstermektedir.

### 3.6.1 Denetimli Öğrenme

Örnekten öğrenme olarak da isimlendirilen denetimli öğrenimde, bir yürütücü tarafından ilgili sınıflar, önceden belirlenen bir kritere göre ayrılarak, her sınıf için çeşitli örnekler verilir. Sistemin amacı verilen örneklerden hareket ederek her bir sınıfa ilişkin özelliklerin bulunması ve bu özelliklerin kural cümleleri ile ifade edilmesidir.

Öğrenme süreci tamamlandığında, tanımlanan kural cümleleri verilen yeni örneklerle uygulanır ve yeni örneklerin hangi sınıfa ait olduğu kurulan model tarafından belirlenir. Örneğin bir program yazılım modeli oluşturuluyor. Modelin çalışmasıyla sonuçların doğruluğuna bakılır. Modelde ilgili değişiklikler yapılarak model değiştiriliyor. Modelde girilen yeni veriler için bir sınıf belirleniyor. Denetimli öğrenmeye sınıflama örnek verilir.

### 3.6.2 Denetimsiz Öğrenme

Denetimsiz öğrenmede, kümeleme analizinde olduğu gibi ilgili örneklerin gözlenmesi ve bu örneklerin özellikleri arasındaki benzerliklerden hareket ederek sınıfların tanımlanması amaçlanmaktadır. Sistem sınıfları bulunmaktadır. Denetimli öğrenmede nesnelere veya örnekler bilinmektedir. Bu sistem, tanımlardan veya örneklerden nesne ve yapıları incelemektedir. Nesnelere üzerinde sınıf tanımları her keşfedilmiş sınıfta kurulmaktadır. Sınıf tanımlarının kümesi denetimsiz öğrenme işlemlerinin sonucudur.(Kaygulu, 1999) Denetimsiz öğrenmeye kümeleme analizi örnek verilir.

Yapay Sinir Ağında öğrenme aşamasında ağ ne kadar eğilirse o kadar doğru sonuca ulaşılır. İki türlü öğrenme var.

Denetimli : Ağa ait hem giriş hemde çıkış değerleri sunulur. Her aşamada örneğe ait çıkış değerleriyle ağ çıkış değeri karşılaştırılarak ağın hatası bulunur. Bu hata kabul edilebilir dereceye indirilene kadar, yapay sinir ağı arasındaki ağırlık değiştirerek iterasyona devam edilir.

Denetimsiz : Bu tür öğrenmede ağı sadece giriş veri grubu sunulur ve ağdan bu veri grubuna uyumlu bir çıkış değeri üretecek şekilde kendisini uygun ağırlıklarla düzenlemesi istenir.

### 3.7 Veri Madenciliğindeki Problemler

- Madencilik işleminde verinin yetersizliği
- Genişlik/Kapsam kavramlarının sınırlı olması
- Büyük boyutluluk
- İlişkilerin nedenlerinin tanımlanması
- Varsayımların olmaması
- Önemli örneklerin sıradan veriye atanması
- İstatistiksel hesapların önemi
- Veri/nitelikler problemini dikkatli incelenmemesi

#### Zorluklar

- Örneklerin anlaşılmaması
- Gerekli veriye ulaşamama
- Veri bütünlüğünü ve işlem modellerini sağlanamaması

### 3.8 Veri Madenciliği Üzerine Yapılan Çalışmalar

DBMiner, Veri Madenciliği sistemi, ilişkisel veritabanlarında ve veri ambarlarında çok seviyeli bilginin madenciliğinde geliştirilmiştir. Bu sistem uygulamaları veri madenciliği fonksiyonları, ardaşık zamanlı örüntüler, sınıflama, tahminleme ve kümeleme içerir. OLAP, istatistiksel analiz gibi veri madenciliği tekniklerinin birleştirilmesi ile iyi performansla etkileşimli veri madenciliği ortamı sağlanmıştır.

Çevrimiçi analitik madencilik(OLAM), çevrimiçi analitik işleme(OLAP) ile bütünlüktür. DBMiner'i diğer VM sistemlerinden ayıran en önemli özelliğidir. Kullanıcılar VT'larına çapraz ve farklı veri altkümelerindeki VM fonksiyonları yerine getirilir. DBMiner'nın gelişmiş veri görselleştirme fonksiyonları kullanıcıya keşfedilen bilgi ve kapsamlı formlardaki sonuçları göstermeye izin verir.

Veri madenciliği sorgulama dili SQL'e benzer DMQL dilidir veya grafiksel kullanıcı arayüzüdür. Kullanıcı DBMiner ile çokboyutsal veri küplerini kolaylıkla

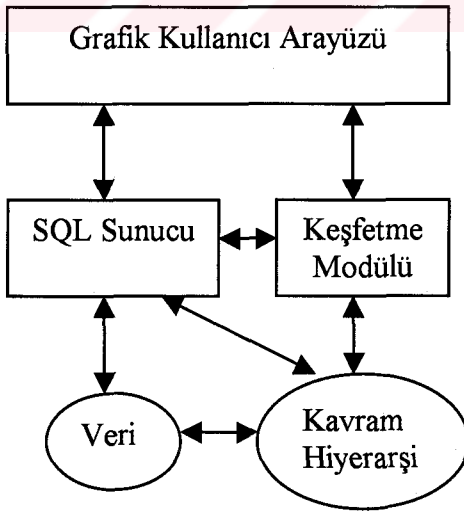
oluşturabilir. Kullanıcı çokboyutsal işlemler olan slicing ve dicing, rolling up ve drill down' ın yapılmasına izin verebilir. DBMiner basit çokboyutsal işlemlerden elde edilen verinin çokboyutsal görünümünü sağlayan görselleştirme araçlarını böylelikle sağlar. DBMiner'in diğer iyi özelliği daha önceden işlenen özetlenmiş veriye sahiptir.

OLTP veri kaynaklarından herhangi veri çıkarma mekanizması sağlanılmaz. DBMiner, kullanıcı oluşturduğu madencilikte verilerin nasıl elde edildiğinden haberdar değildir. Olgu tablosu ambardaki normalleştirilmiş veridir. DBMiner boyut verisini gizlemeyerek farklı boyut tablosu ve nitelik tablosuna gereksinim duymaz. Boyutsal bakış olgu tablolarında tanımlanır ve ilişkisel şemada depolanan tanımlar hakkında bilgilendirir.(Donghua ve ark., 1997)

### Mimari ve İşlevsellik:

DBMiner' ın genel mimarisi şekil 3.5'deki gibi gösterilmiştir. Bilgi keşfi modülleri ve kavram hiyerarşi modülünün bütünleştirilmiş halini bizlere sunmaktadır.

DBMiner veri genelleştirme uygulaması, genelleştirilmiş ilişkiler ve çokboyutsal veri küpü veri yapılarına sahiptir. Genelleştirilmiş ilişki nitelikler kümesi ve toplamlar kümesinden meydana gelir.



Şekil 3.5 DBMiner mimarisi

VM de OLE DB, OLE DB veri sağlayıcısı üzerindeki veri madenciliği işleme-



ri destekleyen bir uzantısı gibidir. Veri madenciliği sağlayıcısı, veri madenciliği algoritmasını sağlayandır. Veri madenciliği müşterisi, veri madenciliği niteliklerini kullanan uygulamalardır. VM'deki OLE DB özelliği Veri madenciliği müşterileri ve veri madenciliği sağlayıcıları arasında API'dir.

VM'de OLE DB yeni sanal nesne oluşturularak veri madenciliği modeli (VMM)'ne gönderilir. VMM bir tabloya çok benzemektedir. CREATE sözdizilimi SQL CREATE TABLE dizilimine çok benzemektedir.

VM'de OLE DB tablonun özel bir tipi gibi davranır. Tablo içerisine veri eklendiğinde VM algoritma ile işlenir ve sonuçlar elde edilir. Sonuç olarak VMM araştırılabilir, refine edilebilir veya tahminlerden türetilir.

Microsoft veri madenciliği çözümü OLE DB'de temel oluşturmaktadır. VM keşif ve analiz olarak kullanılmaktadır. Kurumsal kaynak planlama(ERP), Müşteri ilişki yönetimi(CRM) ve diğer işletimsel sistemlerdeki uygulamalardaki çok büyük veritabanları hesaplanır.

VM'de OLE DB tablonun özel tipi olarak bir veri madenciliği modeli gibi davranır. Tablo içerisine veri yerleştirildiğinde VM algoritması tarafından işletilir ve soyutlama sonuçlanır.

VMM kolonlarının yapısı SQL tablolarına benzer. Modelden kolonlar, giriş verisinden dönüştürülerek elde edilir. Model ile tahminler üretilir. Model kolonları belirli örnek hakkında bilginin hepsini tanımlar. VM'de OLE DB temel işlemler veri madenciliği modeli eğitimidir. Türetme tahminlemelerinde modelin kullanılması izlemektedir. OLE DB veri kaynak nesnesi yaratma ve OLE DB session nesnesi elde etmek. Veri madenciliği model nesnesi yaratmaya ait sorgular,

```
CREATE OLAP MINING MODEL <Model Name> FROM <Case Cube Name>
(<Cube Members>) USING <Algorithm Name>
```

```
CREATE OLAP MINING MODEL [MyOlapModel] FROM [Sales]
```

```
(
```

```
  CASE
```

```
    DIMENSION [Customers]
```

```
    LEVEL [Name]
```

```
    PROPERTY [Marital Status],
```

PROPERTY [Education],  
PROPERTY [Member Card Type] PREDICT

)

USING Microsoft\_Decision\_Trees

### 3.9 Mevcut Veritabanlarında Analiz Servisleri ile Veri Hazırlama ve Veri Madenciliği

Tanımlanan bir problemin çözümünde, verinin temizlenmesi ve hazırlanması, veri kümesinde veri araştırması, VM modelinde uygun kolonların seçimi ve son olarak VM modelleri karşılaştırılır.

#### Veri Madenciliği Çözümü

- Problem Tanımlama,
- Veri Hazırlama,
- Modellerin Yapımı,
- Modellerin Onaylanması,
- Modellerin Yerleştirilmesi.

Kısaca veri madenciliği verinin temizlemesi ve dönüşümü ile meta verinin ilişkilendirilmesinin yönetim ve modellerin yapımı ve yerleştirilmesinin yapılabilmesidir.

#### Analiz Servisleri:

Veri madenciliği modelleri yaratmada ve depolamada kullanılır ve modellerin performansı test etmede ve tahminlerin yaratılmasında kullanılır.

Analiz servisleri ilişkili ve çokboyutsal veri kaynaklarından veri madenciliğini destekler. Analiz servisleri algoritmaları herhangi bir ilişkiyel veri kaynağından veri ile veri madenciliği modeli geliştirilir. OLE DB erişimin sağlanması ile Analiz servisleri ile çokboyutsal küpler yaratılır.

Microsoft veri madenciliği çözümü VM'de OLE DB üzerine esaslanılmıştır. VM keşif ve analiz olarak kullanılır. ERP, CRM ve diğer işletimsel sistemlerdeki uygulamalardaki çok büyük veritabanları hesaplanır.

SQL Server 2000 iki veri madenciliği algoritması Microsoft Karar Ağaçları ve Microsoft Kümelemedir. SQL Server 2000 analiz servislerinin iki bileşeni OLAP ve VM'dir. Veri analizinde kullanılan iki önemli tekniklerdir.

### Veri Madenciliği Modeli:

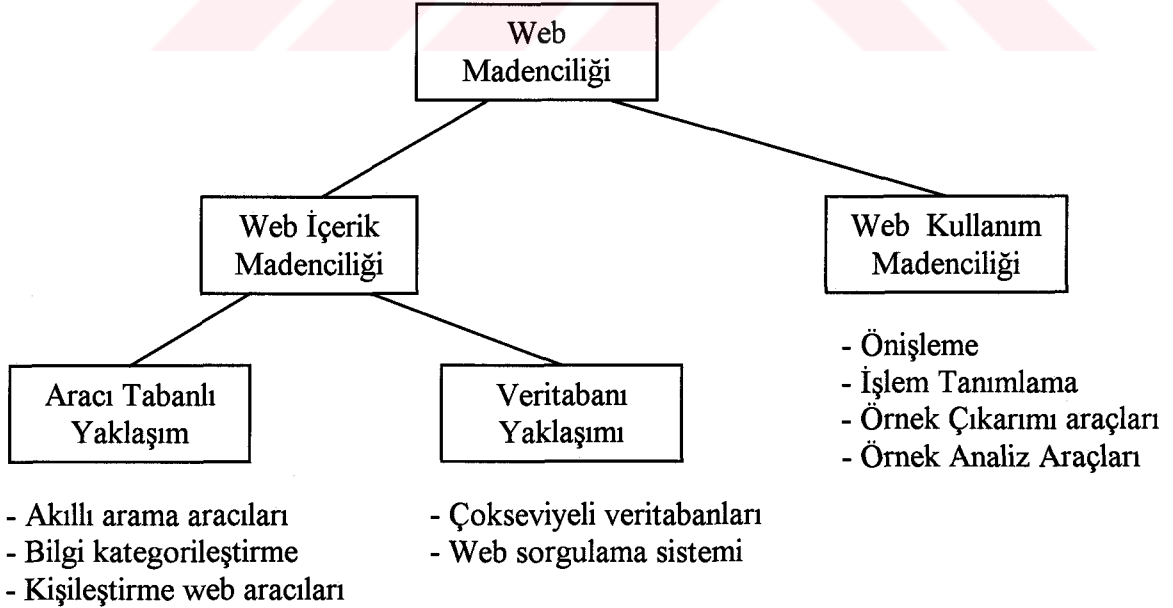
VM modeli ilişkisel tablo gibi düşünülebilir(farklı veri tipleriyle kolonların listesini içeren tablo gibi) Buradaki kolonların bazıları giriş kolonları ve bazıları tahmin edilebilir kolonlardır. Aşağıdaki örnekte müşteri bilgisindeki risk seviyesine ait örnek verilmiştir. Microsoft Karar Ağaçları algoritması kullanılır.

SQL Server 2000 SQL Server, Access ve Oracle' daki OLE DB sürücüleri ile ilişkilidir. Eğitim işlemleri bu aşamada bazı zamanlar işlenebilir.

### 3.10 World Wide Web'te Veri Madenciliği

İnternet üzerinde bilgilerin artışıyla bazı otomatik araçlar kullanılmaya başlanmıştır. Sunucu tarafındaki gelişimlerde bilgi bulma gereksinimleri ve madencilikteki istemci tarafındaki zeka sistemleri gelişim gereksinimlerinde bilgi bulma önemli kullanılır. Web madenciliği, World Wide Web(WWW)'dan yararlı bilgi analizi ve keşfinde tanımlıdır. Web içerik madenciliği ve kullanıcı erişim örneklerin keşfi web kullanım madenciliğidir.

Web madenciliği sınıflandırılması şekil 3.6'de gösterilir.



Şekil 3.6 Web Madenciliği sınıflandırılması

Web içerik madenciliği, WWW' deki düzensiz yapıdaki bilgi kaynaklarından web tabanlı bilginin otomatik keşfini sağlar. Yapısal bilgi ve sınıflamayı, bilgi süzme veya yorumlama sağlamayabilir. Bu faktör birçok araştırmacıya bilgiyi yeniden elde etmek için daha akıllı araçlar kullanmaya neden olur.

Web kullanım madenciliği, Web sunucudan kullanıcı erişim örneklerinin otomatik keşfmekte işlemidir. Organizasyonlar, Web sunucuları tarafından üretilen ve sunucu erişim loglar(kütük)da toplanmış günlük büyük değerdeki verileri toplar.(Ramakrishnan, 1997)



## 4. ÜNİVERSİTE BİLGİ SİSTEMİ İÇİN VERİ AMBARININ OLUŞTURULMASI İLKELERİ VE ÖRNEK UYGULAMALAR

Bu bölümde, önceki bölümlerde anlatılan kuramsal bilgiler veritabanı, veri ambarları ve veri madenciliği uygulaması üzerinde gösterilmiştir. “Personel Bilgi Sistemi” (PBS) olarak adlandırılan bu uygulama çalışmamızın amacına uygun olarak başlangıçta birden fazla Nesneye yönelik, ilişkisel bir veritabanı uygulaması oluşturulmuştur. Ve 2. bölümde anlatılan Veri Ambarı uygulamasının yapılması ile Üniversite Veri Ambarı oluşturulmuştur. Ve 3. bölümde anlatılan Veri Madenciliği ile gelecek dönemlerdeki tahminlerin ve analizlerin uygulanması ele alınabilir.

Veri ambarı uygulaması başlık 2.10’da listelenen yıldız şema ve veri küpleri teknolojileri uygulanarak oluşturulmuştur.

Veri ambarları oluşumunda kullandığımız modeller bizlere veriler üzerindeki görünümde kolaylık sağlamıştır. Yıldız şema, veritabanları arasında karmaşık ilişkinin yerine kar tanesine benzeyen bir yıldız şema olarak birden fazla boyut tabloları ve bir olgu tablosunun oluşturduğu bir model sağlamaktadır. Veri küpleri ise bize özetleme ve genelleme gibi işlemleri uygulayarak veriler üzerinde veri küpü işlemlerinin yapılmasını sağlamaktadır.

Tüm bu özellikleri içermek üzere, PBS tasarlayarak, akademik personele ait üniversite , fakülte ve bölüm sınıflandırılması üzerinde uygulaması gerçekleştirilecektir.

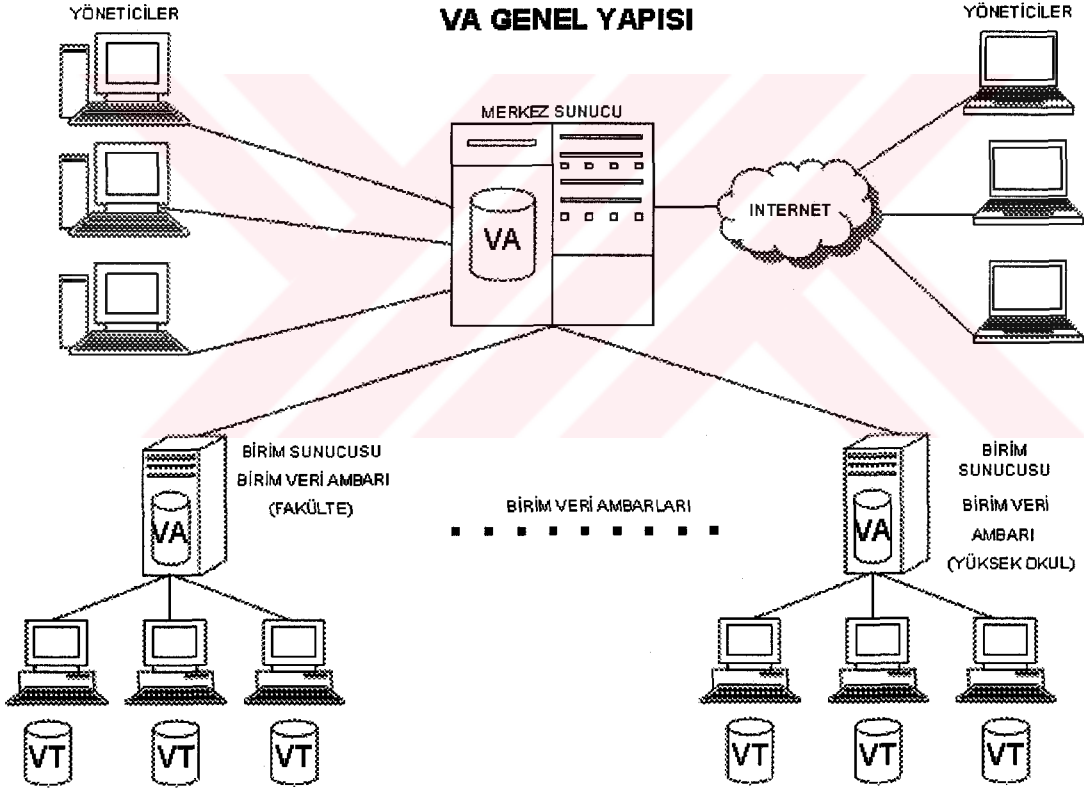
Oluşturulan karar destek sisteminde, birden fazla birim üzerinde oluşan bilgi kaynaklarının kapasitesi çok büyüktür. Bu kaynaklardan boyutlar indirgenerek istenilen bilgiye daha kolay ve daha hızlı erişmemizi sağlamaktadır. Ve yıllar üzerinde verinin dağıtımı bizlere elde ettiğimiz bilgiler doğrultusunda geleceğe yönelik tahminlerde bulunmamızı sağlamaktadır.

### 4.1. Problemin Tanımı

Personel Bilgi Sistemi (PBS)’nin diğer sistemlerden farkını oluşturulan işlemleri genel olarak açıklayalım. Üniversite Bilgi Yönetim Sistemini oluşturan çok sayıda işlevsel veritabanları üzerinde işlemlerin daha verimli yapılması için Veri Ambarlarına gerek duyulur.

Üniversite'nin dağıtık bilgi sistemi üniversite – fakülte – bölüm hiyerarşisinden oluşmaktadır. Bu hiyerarşideki alanlarının hangi nesnelere oluştuğuna bakalım.

Bu bilgi sisteminde üniversite sınıfı fakülte nesne özelliklerinin genelleşmesinden oluşmuştur. Herbir fakülte nesnesi ise uygun bölüm nesnesinin genelleşmesidir. Herbir bölüm nesnesi ise fakülte boyutsal tablosunun bir satırında tanımlanır. Her ölçüm değeri, hiyerarşinin alt seviyesindeki olgu tablolarından uygun hesaplama değerlerinin hesaplanmasından elde edilebilir. Bu yapıyı göz önünde bulundurarak Üniversite Bilgi Yönetim Sistemi için hiyerarşik yapıya veri ambarı mimarisi önerilmiştir. Şekil 4.1'de Üniversite Veri Ambarının genel yapısı gösterilmektedir. (Salahlı, 2003)



Şekil 4.1 Üniversite Veri Ambarının genel yapısı

Üniversitelerde yerleşke dağıtım, bilgi ambarı sisteminde uygun mimarinin seçilmesini gerektirmektedir. Böyle sistem bilgisayar ağları üzerinde dağıtık alt sistemlerin bütünleşmesi gibi düşünülebilir. Fakülteler üzerindeki veri ambarlarının oluşmasında bölümler veritabanlarındaki verilerin üzerinde veri ambarı işlemleri

gerçeklenerek fakülte veri ambarı oluşturulur. Bir üniversite mimarisi birden fazla fakülteler, yüksekokullar, meslek yüksekokullar gibi alt birimlerden oluşmaktadır.

Veri ambarındaki amaç, verilerden karar destek sağlayan bilgiler elde etmektir. Fakülte veri ambarından fakültedeki yöneticilerin istedikleri raporlar ve özetlemeler gerçekleştirilir. Ayrıca zaman hiyerarşisine göre bir üst birimlerin istedikleri analizlere kolaylıkla erişim sağlanır. Açıkçası her birimde oluşturulan veri ambarı tek bir merkezi veri ambarına bağlanmaktadır. Bu merkezi veri ambarında üst düzeydeki yöneticilerin istedikleri raporlar ve işlemler gerçekleşmektedir. (Yüksek ve ark., 2003)

Yukarıdaki Şekil 4.1'deki şemaya göre en alt kademedeki VT'ları Personel Veritabanı, Fakülte Veritabanı, Öğrenci Veritabanı örnek olarak verilebilir. Personel Veritabanından,

```
SELECT COUNT(PSicilno)
FROM Personel
WHERE birimadi=" Müh.-Mim."
```

Birim veri ambarı olarak Fakülte veri ambarına gerekli zaman aralıkları ile yıl, ay, personel şeklinde kayıtlar eklenir.

```
MMFVeriAmbarı(BirimKodu, Yıl, Ay, Per_Sayı)
FEFVeriAmbarı(BirimKodu, Yıl, Ay, Per_Sayı)
```

Şekil 4.1'deki üst katman olarak merkezi veri ambarına verilerin çekilme işlemi Fakülte veri ambarlarından Mühendislik-Mimarlık Fakülte veri ambarı, Fen-Edebiyat Fakültesi veri ambarı ve bununla birlikte diğer birimlere ait veri ambarlarından veriler çekilir. Bunun sonucunda aşağıdaki tablolar oluşmaktadır.

```
SELECT SUM(Per_Sayı)
FROM MMFVeriAmbarı, FEFVeriAmbarı
WHERE Yıl="2002" And Ay="2"
```

Sorguları işlenerek Merkezi Veri Ambarı oluşmaktadır.

Yıl, Ay, Toplam\_Per alanlarını kapsayan tablo oluşturulur.

```
SELECT SUM(Per_Sayı)
FROM MMFVeriAmbarı, FEFVeriAmbarı,...
WHERE Yıl="2002"
```

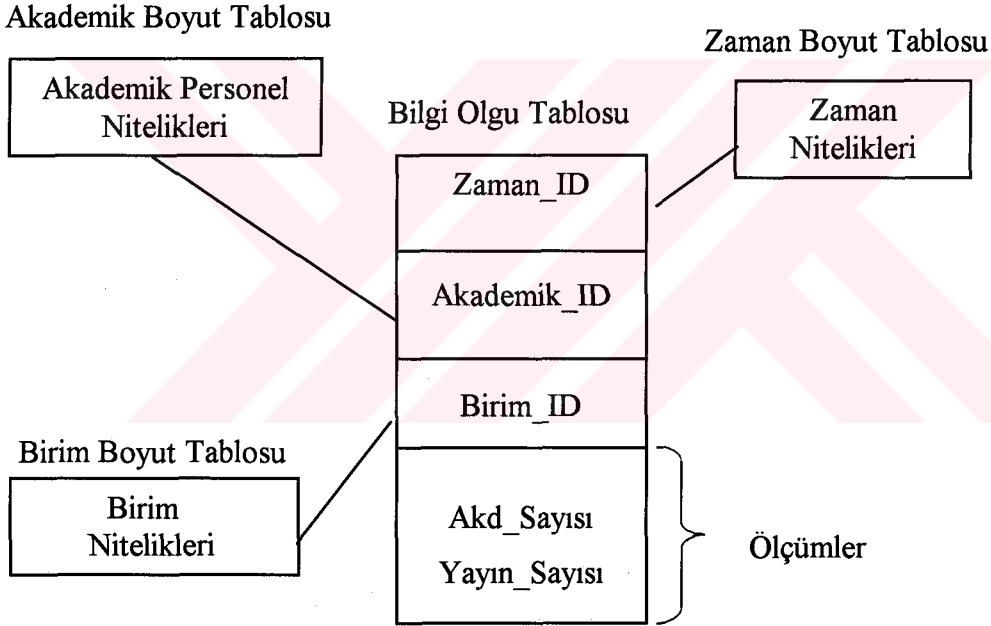
Yıl, Toplam\_Per alanlarını kapsayan tablo oluşturulur.

## 4.2 Şemaların Tasarımı

Kaynaklardan sonuçlar çıkararak karar desteğe yönelik bilgiler toplanılması ile yöneticilerin karar vermesine yardımcı olacak veriye ulaşmasına, verinin özetlemesine ve analiz edilmesine yardımcı olmaktadır.

Şimdi bu özellikleri taşıyan veri ambarının oluşturulması, karar destek sistemleri için tasarlanan mimari ve bu problemlerin çözümlerinin uygulanması aşağıdaki gibidir.

Üniversite bilgi sistemi veri ambarının mantıksal modellemesi yapılarak şema örnekleri oluşturulmuştur. Şema oluşumunda özellikler Birim, Akademik, Zaman'dan oluşan boyutlardır



Şekil 4.2 PBS'nin Mantıksal Modeli (Özgün)

Bu yıldız şemayı bir sorgu üzerinde gerçekleştirelim. Müh-Mim. Fakültesindeki Prof., Doç. Akademik ünvana sahip 2000-2001-2002 yılları arasındaki toplam sayı nedir.

Uygulamadaki Üniversite\_Olgu tablosunun alanları:

Üniversite\_Olgu(Unvan\_Kodu, Zaman\_Kodu, Birim\_kodu, Toplam\_Sayı)



Bu olgu tablosunun oluşması için Unvanlar, Zaman, Birimler boyut tablolarından veriler çekilmiştir. Veriler üzerinde istenilen fonksiyonlar uygulanarak hesaplama değerleri elde edilir.

Mantıksal modelde tek bir olgu tablosu vardır. Olgu tablosu boyut tablolarına ait anahtarlar ile ilişkilendirilmiştir.

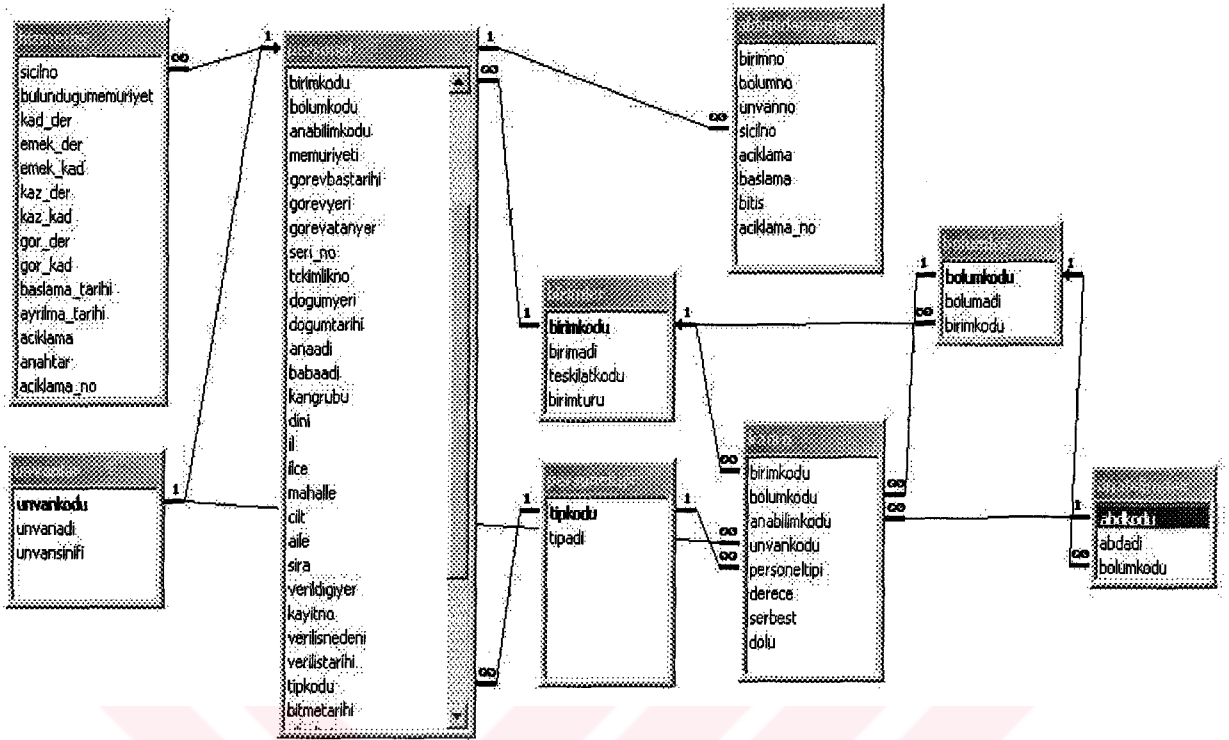
Buradaki ilişkiler OLAP uygulamalarında olgu tablosundan daha kısadır. Boyut tablosu olarak isimlendirilir. OLAP sistemleri bütün bilgiyi depolayan, olgu tabloları içerir, ilişkiler ROLAP sistemler gibi adlandırılır.

Mimaride kullanılan donanım, merkezde güçlü bir sunucudan başlayarak, birim sunucuları, ve daha alt seviyedeki sunucularla meydana getirilmiş hiyerarşik bir yapı oluşturmaktadır.

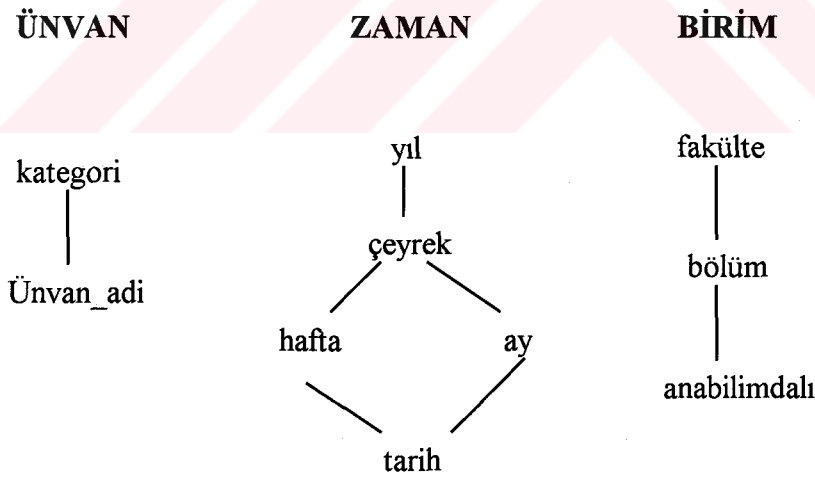
Mimarinin en alt seviyesinde kullanıcıların görsel programlama dilleri ile yazılmış programları kullanarak işlemlerini gerçekleştirdikleri işlevsel veri tabanlarına Personel Bilgi Sistemi, Öğrenci Bilgi Sistemi, Fakülte Bilgi Sistemi örnek olarak verilebilir. Bu veri tabanları birim sunucusu üzerinde çalışan ortak veri tabanları da olabilir. Birim sunucuları aynı zamanda Birim Veri Ambarını da barındırmaktadır. Birim veri ambarları için orta ya da küçük boyutlu veri tabanı yönetim sistemlerinden birisi kullanılır. Merkezi sunucuda ise güçlü bir veri tabanı yönetim sistemi bulunur. Yöneticiler daha genellenmiş verilerin bulunduğu merkezi sunucuda bulunan veri ambarını kullanarak istedikleri bilgilere ulaşırlar. Merkez sunucuya internetten de erişilebilir olması yöneticilerin dış ortamlardan da bu verilere ulaşmasını sağlar.

Üniversite Bilgi Sistemi veri Ambarı oluştururken Şekil 4.1'deki yapıda Personel VT bileşeninin bir bölümünü tablolara halinde gösterimi Şekil 4.3' de verilmiştir.

Bu kullandığımız sistemlerden sadece biridir. Bununla birlikte istenirse yeni sistemlerde ilave edilerek oluşturulan mimari büyütülür. Bu tablolardaki veriler üzerinde sınıflamaya ait bir hiyerarşi oluşturulur. Hiyerarşi örneği şekil 2.7' de gösterilmektedir.



Şekil 4.3 Personel Veri Tabanının bir bölümü üzerindeki tablolara ait ilişkiler



Şekil 4.4 Boyutsal Hiyerarşi

Bu hiyerarşi yapısı Ünvan bir dereceye kadar ayrıştırılır. Profesör, Doçent, Yardımcı Doçent, Araştırma Görevlisi, Öğretim Görevlisi, Uzman şeklinde oluşur. Zaman yıllardan başlayarak gün değerine kadar ayrıştırılabilir. Birimler ise Üniversite, fakülteler, bölüm, anabilimdalı şeklinde ayrıştırılır.

### 4.3 Raporların Oluşturulması

Son kullanıcı seviyesinde oluşturduğumuz raporları inceleyelim. Yukarıdaki veritabanlarından veri ambarı oluşturduk OLAP ve MOLAP işlemlerinden geçirildikten sonra elde edilen son kullanıcı arayüzleri ile birlikte raporlar incelenmektedir.

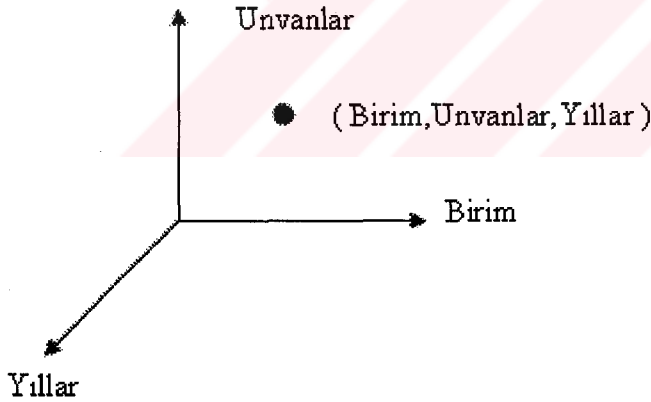
Birincisi giriş sayfasından istenilen raporun seçilmesidir. Birinci raporda,

Birimlerde öğretim elemanlarının akademik ünvana göre kadro durumunun değerlendirilmesi:

Kadro durumunun akademik ünvana göre mevcut kaç kişi olduğunu ve bunun yıllar üzerinde dağılımı elde edilmesi uygulamasına bakılır.

Kadro veritabanında yeni bir eleman eklenmesi gerektiğinde sayı bir artırılır ve bir önceki sayı kaybolur. İşte bu sayının yıllar üzerinde saklanmasıyla veri ambarları oluşturulmaktadır. Bu oluşturulmada genelleme ve ayrıntılarına ayırma, gruplaştırma işlemlerinin yapılması ile sağlanılmaktadır.

Bu raporun oluşumunda Unvan, Birim, Yıllar parametreleri kullanılır.



Şekil 4.5 Birim, ünvan ve yıl parametrelerinin koordinat ekseninde gösterimi

Bu üç boyutlu ekseninde belirlenen noktalar bizim belirlediğimiz parametrelerin programda işlenmesi sonucu oluşur. Programda bu hiyerarşiye ait birimleri hangi bileşenlerin ifade ettiğine bakalım.

Birimler ağaç yapısındadır.

```
For i = 1 To BirimTree.Nodes.Count
    If ( BirimTree.Nodes(i).Selected = True ) Then
        .....
    End If
Next i
```

Ünvanlar List halindedir.

```
For i = 0 To UnvanList.ListCount - 1
    If UnvanList.Selected(i) = True Then
        .....
    End If
Next i
```

Yıllar boyutu ise checkbox halinde hangi yılların seçildiği kontrol edilir.

```
For i = 0 To YılSayısı
    If YılCheck(i).Value = 1 Then
        .....
    End If
Next i
```

Seçilen parametreler

```
Ambar_Olusturma( tablo_adi, unvan_kodu, gorev_tarihi, birim_adi )
```

fonksiyonuna gönderilir.

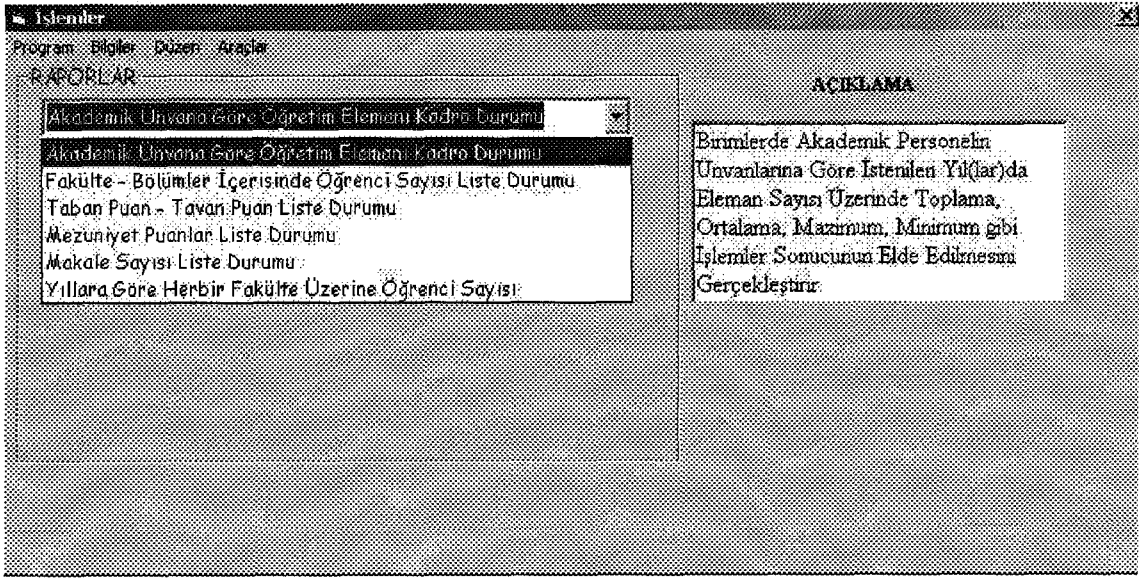
Bu fonksiyonda girilen parametrelerin ait olduğu tablodan veri çıkarımı yapılır. Elde edilen veri üzerinde özetleme fonksiyonları toplam, ortalama, maximum, minimum ve sayısını bulma fonksiyonlardan istenilen uygulanarak bilgi çıkarımı gerçekleşir. (Karagülle ve ark., 1999)

```
Grafik_Cizimi( alan1_Sayı, alan2_Sayı, alan1, alan2, boyut );
```

Grafikte satır ve kolon değerleri her rapora göre değişiklik gösterdiği için fonksiyona gönderilen parametreler içerir değişir. Örneğin ilk rapora göre;

```
Grafik_Cizimi( unvan_Sayı, birim_Sayı, unvanboyutu, birimboyutu, boyut );
```

Elde ettiğimiz bilginin son kullanıcıya daha kolay anlaşılmasını sağlamak için grafik ortamına aktarılır. Program çalışmasında daha önceden belirlenen raporlara ulaşmayı ve bu raporun işlevinin ne olduğuna dair açıklama sağ tarafında Açıklama kısmında verilmiştir.



Şekil 4.6 Rapor Seçimi

Raporumuzda özetleme fonksiyonu olarak toplam, ortalama, maximum değerler ve minimum değerlerinin hesaplanma işlemleridir. İlk rapor için; istenilen özellikler yani seçilen parametrelere göre toplam kişi sayısı, ortalama kişi sayısı ve ağaç yapısında seçilen birim yaprak değil de düğüm ise maximum sayıdaki birimler veya minimum değerindeki birimler isimlerinin öğrenilmesi işlemleridir. Bu işlemler özetleme fonksiyonlarıdır.

Birinci raporu seçtiğimizde şekil 4.7'deki arayüz kullanılmıştır. Birimlere ait ağaç yapısını, yukarıdan aşağıya doğru ayırıştırmak mevcuttur. Mühendislik Mimarlık Fakültesi düğümüne tıklanıldığından bu birimin altındaki bölümler açılmaktadır. Bilgisayar Mühendisliği yapaktır. Ünvanlar listesinden seçimimiz bir veya birden fazla ünvan adına göre olabilir. Yılları checkbox şeklinde klik işlemiyle istenilen yıllar üzerinde sorgulama yapabiliriz.

```
SQL = "select * from Unvanlar"
```

```
Set rec = db1.OpenRecordset(SQL, dbOpenDynaset)
```

```
While Not rec.EOF
```

```
    If (rec.Fields("unvansinifi")="ÖE") Then List2.AddItem rec.Fields("unvanadi")
```

```
rec.MoveNext
```

```
Wend
```

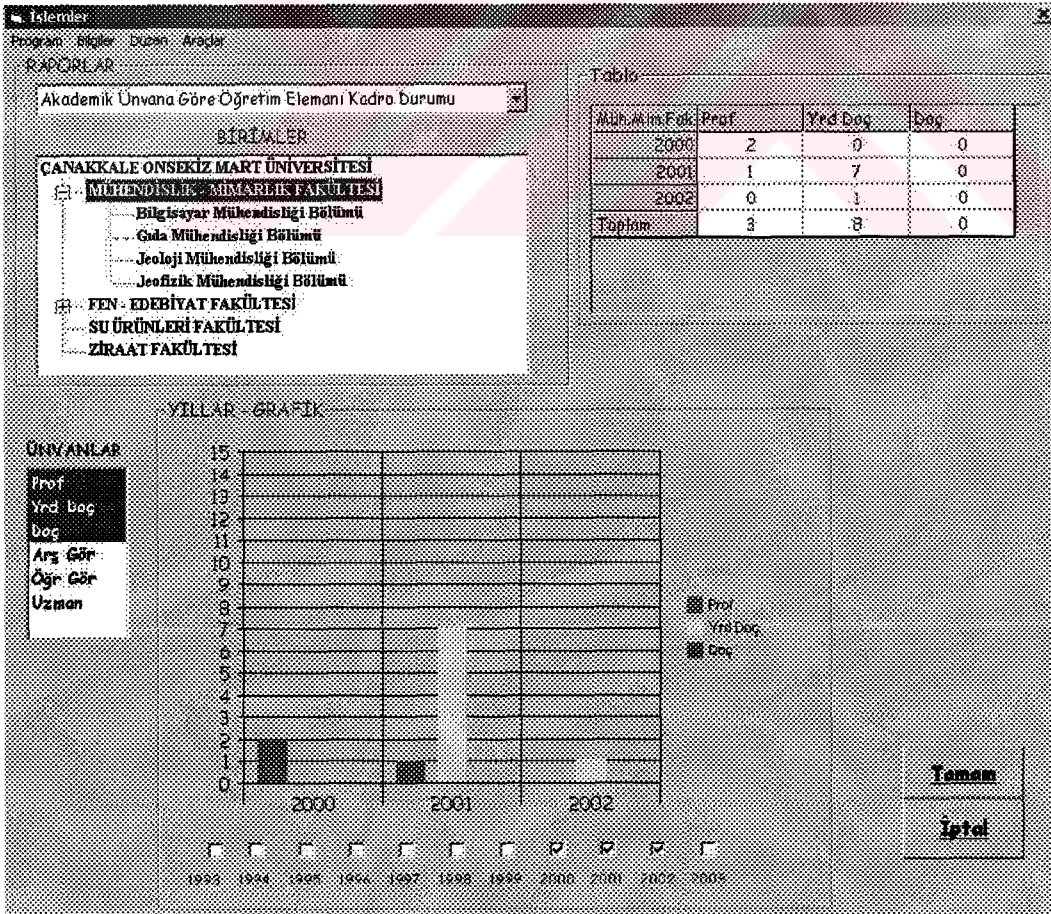
SQL sorgusu ile ünvan listesi oluşmuştur. Birimler ağaç yapısının oluşumu;

```

Call Tree1(0).Nodes.Add( , "fak", "ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ")
Call Tree1(0).Nodes.Add("fak", tvwChild, "mmf", "MÜHENDİSLİK - MİMARLIK FAKÜLTESİ")
Call Tree1(0).Nodes.Add("fak", tvwChild, "fenedb", "FEN - EDEBİYAT FAKÜLTESİ")
Call Tree1(0).Nodes.Add("fak", tvwChild, "suUrun", "SU ÜRÜNLERİ FAKÜLTESİ")
Call Tree1(0).Nodes.Add("fak", tvwChild, "ziraat", "ZİRAAT FAKÜLTESİ")
Call Tree1(0).Nodes.Add("mmf", tvwChild, "bol1", "Bilgisayar Mühendisliği Bölümü")
Call Tree1(0).Nodes.Add("mmf", tvwChild, "bol2", "Gıda Mühendisliği Bölümü")
Call Tree1(0).Nodes.Add("mmf", tvwChild, "bol3", "Jeoloji Mühendisliği Bölümü")
Call Tree1(0).Nodes.Add("mmf", tvwChild, "bol4", "Jeofizik Mühendisliği Bölümü")

```

verilmiştir. Bu örnek sadece Mühendislik Mimarlık Fakültesine ait alt birimleri kapsamaktadır. Yukarıda anlattıklarımızın arayüzü şekil 4.7’de verilmiştir.



Şekil 4.7 Akademik Ünvana göre öğretim elemanı kadro sayısı

Boyutlardaki veriler üzerinde istenilen işlemlerin gerçekleştirilmesini sağlayan toplama işlemleri histogramı, çapraz-tablolama, üç veya daha fazla parametremiz varsa roll-up, toplamların bulunması, drill-down ise alt toplamların bulunması ve alt toplama yapıları kullanılır. Bu işlemlere daha önceki teorik bölümlerde örnekler verilmiştir.

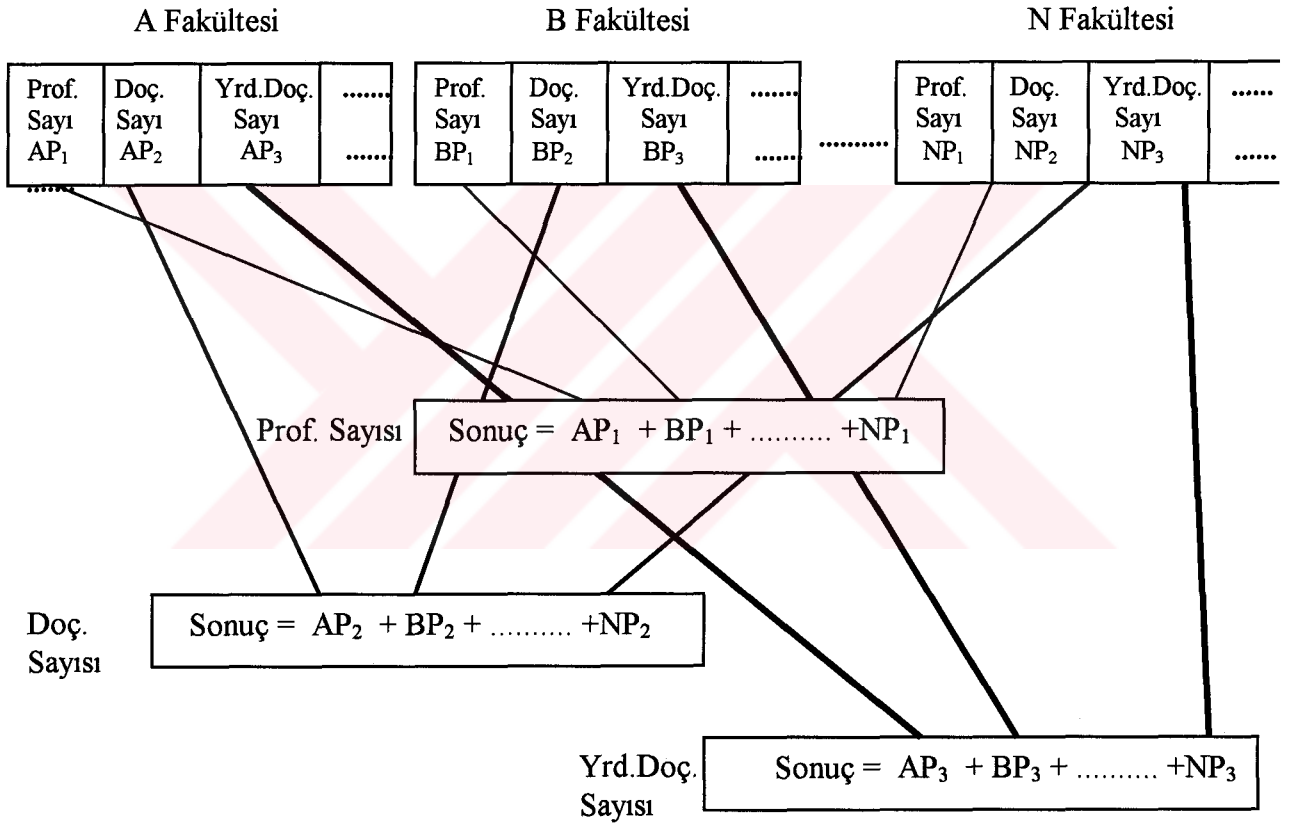
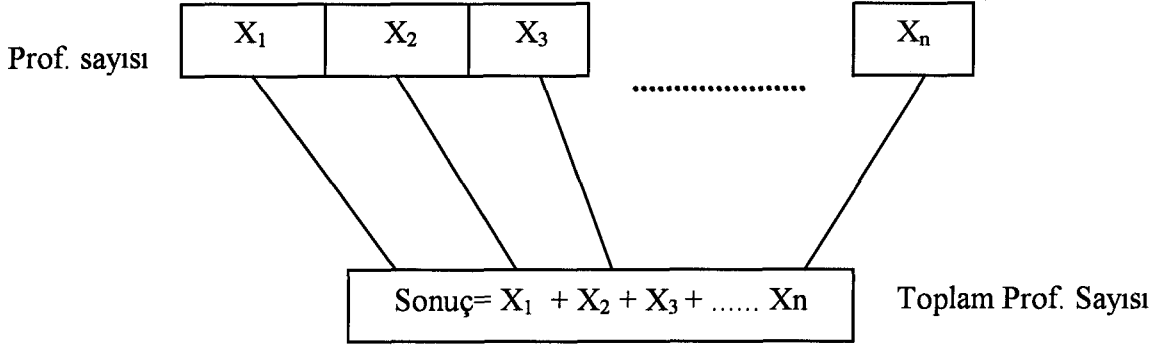
Verilerin çokboyutlu gösterimi  $n$  uzay boyutu gibi  $n$  kümeleme niteliklerinin davranışı gibi görülmektedir. İki boyutlu çapraz tablolama ve üç boyutlu olarak küp şeklinde gösterime örnekler verilebilir.

Üç giriş parametresine sahip veri küpü oluşturulmasını incelediğimizde aşağıdaki adımlar yapılmaktadır.

Toplam değer bize parametrelere ait sorgulama sonucunda elde ettiğimiz toplam değerdir. Toplam akademik sayısını, toplam öğrenci sayısını, toplam makale sayısını verebilir.

Bu işlem alanı bize sonuç hesaplama değerini verir. “GROUP BY” işlemi ile istenilen alanlarda veriyi gruplayarak işlemi bunun üzerinde gerçekleştiririz. Bu işlem bir örnek üzerindeki uygulaması verilmiştir.

### Fakülteler



Yukarıdaki örnekteki toplam değeri boyutlar bazında yapılan işlemler sonucundaki toplam değeridir. Herbir akademik ünvana göre öğretim elemanın toplam kadro sayısını hesaplayabiliriz.

Çapraz tablo olarak tek tek yıllar üzerinde iki parametreye göre çapraz tablolama oluşturulur.



İki boyut olarak ünvanlar ve birimler üzerinde “GROUP BY Birim, Ünvan” işlemi gerçekleştirilerek yukarıdaki çapraz tablo oluşturulur.

Çizelge 4.1 Çapraz Tablo gösterimi

	Müh.Mim.	Fen-Edb.	Ünvan göre Grublama	
Prof.				
Doç.				
Yrd.Doç.				
Birim göre Grublama				ToplamDeğer

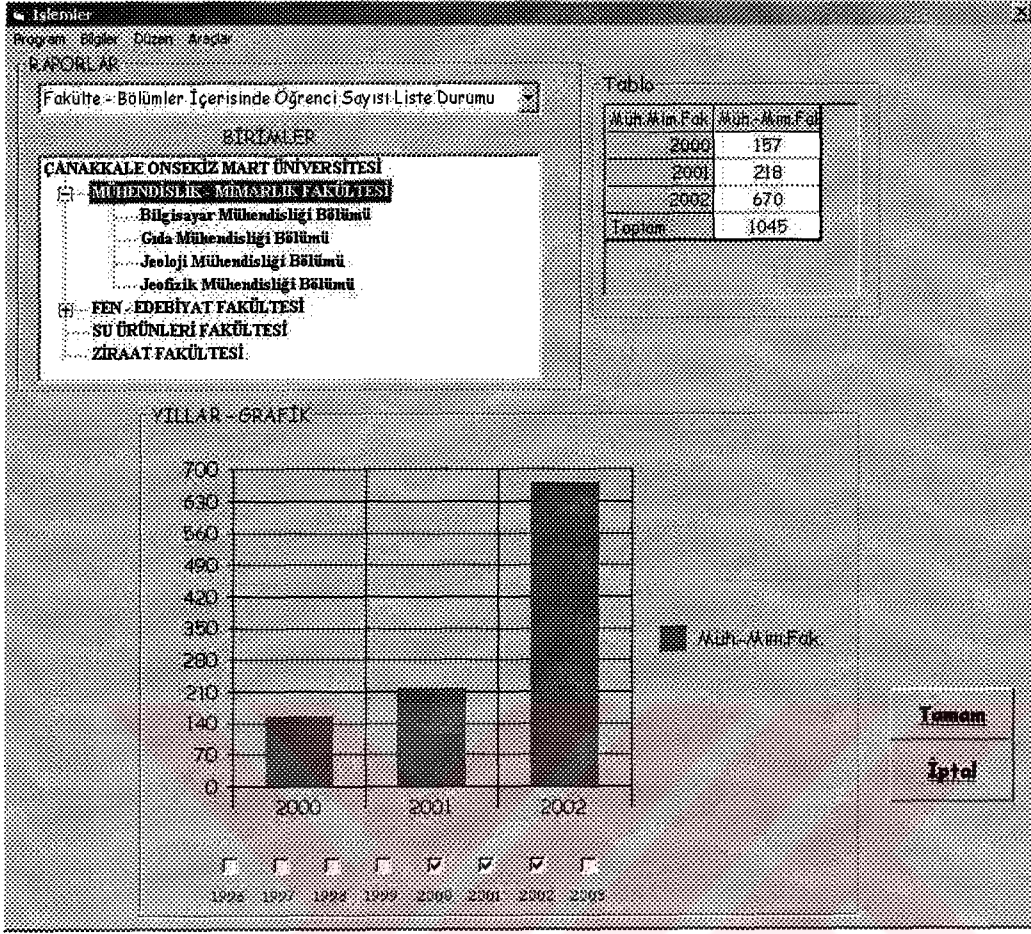
Bu işlemler sırasıyla yapıldıktan sonra işlemlerimizi küpün yan alanları üzerinde dağıttığımız işlemlerimiz kolaylıkla elde edilir. Küpe ait kenarlar Birim, Akademik Ünvan ve Zaman'dan oluşmaktadır. Küpe ait şekil 2.4'te gösterimi yapılmıştır.

Birimlerde öğrenci sayısının hesaplanması:

Herbir bölümdeki öğrenci sayısının hesaplanması ile fakülte altındaki her bir bölüm öğrenci sayısı üniversite genelinde öğrenci sayısı hesaplanmaktadır. Öğrenci sayısının yıllar üzerindeki değişiminin hesaplanması ile artış veya azalış oranları elde edilmektedir.

Öğrenci sayısının önceki yıllar ile karşılaştırılması sonunda öğrenci sayısı artışının hangi birimlerde olduğunun gösterimi hesaplanır. Öğrenci işlerinde mevcut öğrenci sayısını bildiren çizelgeler her ay rektörlük birimine verilmektedir. Bu rapor sonucu elde edilen grafikten istenilen tarihler arasında veri istatistiğine ulaşılabilir.

İkinci raporumuzda, birimler genelinde zaman boyut hiyerarşisinde yıllar sınıfını üzerine toplam fonksiyonu işlenmiştir. Bu parametrelere göre arayüz şekil 4.8'de gösterilmiştir.

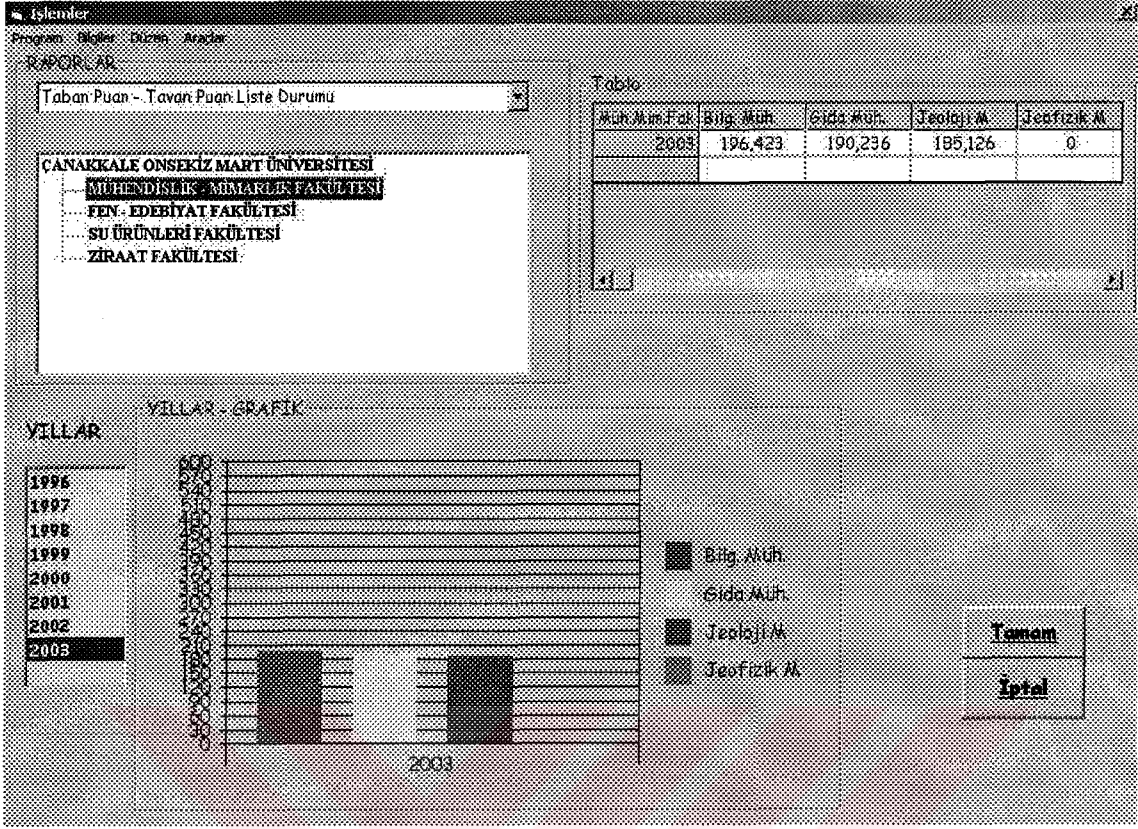


Şekil 4.8 Fakülte-Birimler içerisinde öğrenci sayısı

Bölüme giriş puanı olan taban puanları hesaplanması:

Öğrencilerin üniversite sınavı sonucunda kazandıkları bölümlerin giriş puanı olan taban puanları üzerinde ayrıntıların hesaplanmasıdır.

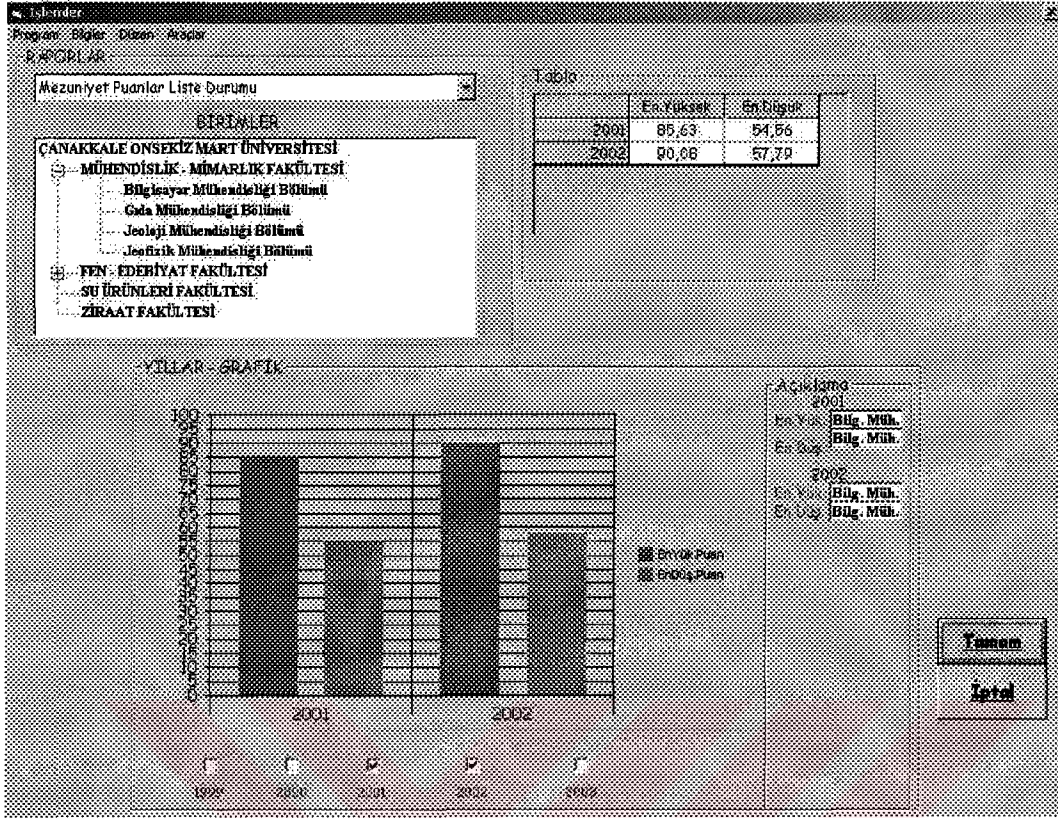
Üniversite genelinde en yüksek puana sahip bölüm adı ve öğrenci numarası sorgulaması yapılabilmektedir. Ayrıca önceki yıllar üzerindeki değişim analizleri de elde edilmektedir. Fakülteadaki bölümler arasında veya üniversite genelinde en yüksek puana sahip bölüm ve en düşük puana sahip bölümün hangisi olduğunu öğrenebiliriz. Bu raporu daha farklı şekilde genişletebiliriz. Giriş puanlarına ait tablo ile öğrenciler ait tablo arasındaki işlemler sonucunda bölüme en yüksek puan ve en düşük puanla giriş yapan öğrenci hakkında bilgi sahibi olabiliriz. Bu taban puanlarına ait arayüzü şekil 4.9'da gösterimi yapılmıştır.



Şekil 4.9 Giriş Puanlarının-Taban Puanlarının Listelenmesi

Mezuniyette bölümde öğrencilerin ders ortalamasının sonucunda diploma notların hesaplanması:

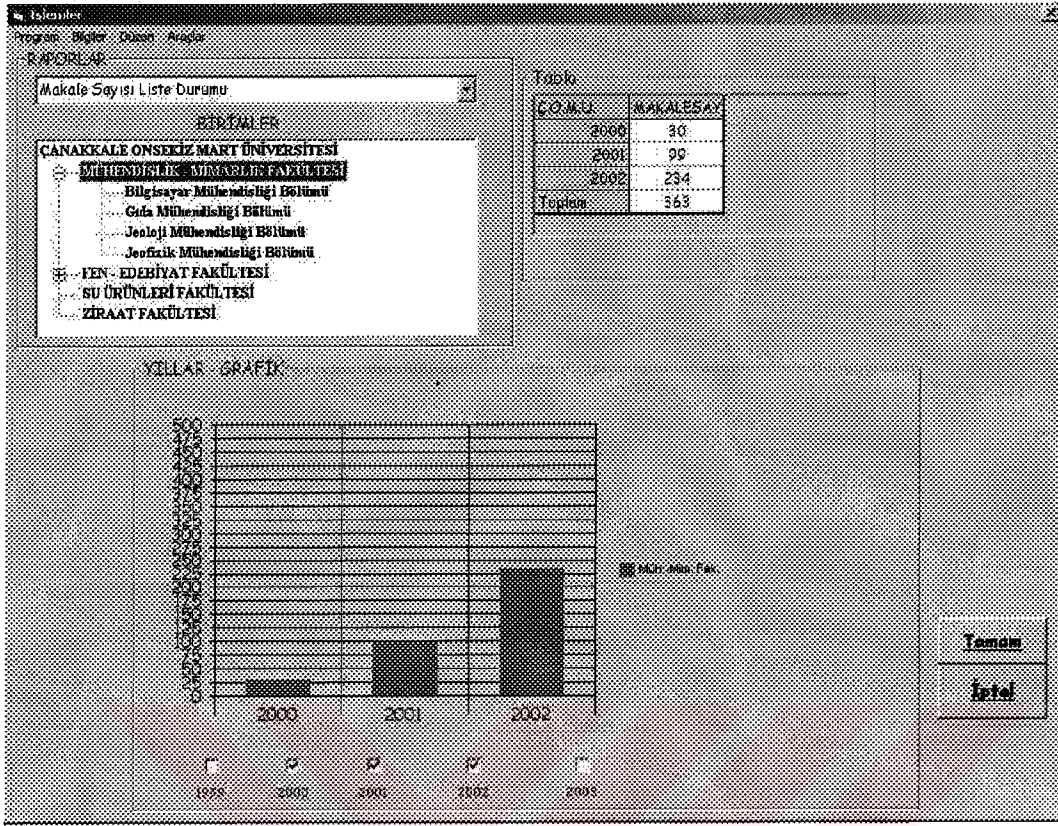
Bölümdeki diploma notlarına göre öğrencilerin başarı yüzdelerine göre en yüksek, en düşük not veya sınıf içerisindeki başarı ortalamasının hesaplanması elde edilir. Bölümden mezun olan öğrenci sayısı ile birlikte öğrencilerin en düşük ve en yüksek puan değerlerini minimum ve maximum toplam fonksiyonları hesaplanmaktadır. Böylelikle bölümün önceki yıllardaki başarı oranları karşılaştırılır. Buna ait arayüz şekil 4.10'de verilmiştir.



Şekil 4.10 Mühendislik-Mimarlık Fakültesine Ait Mezuniyet Puan Listesi

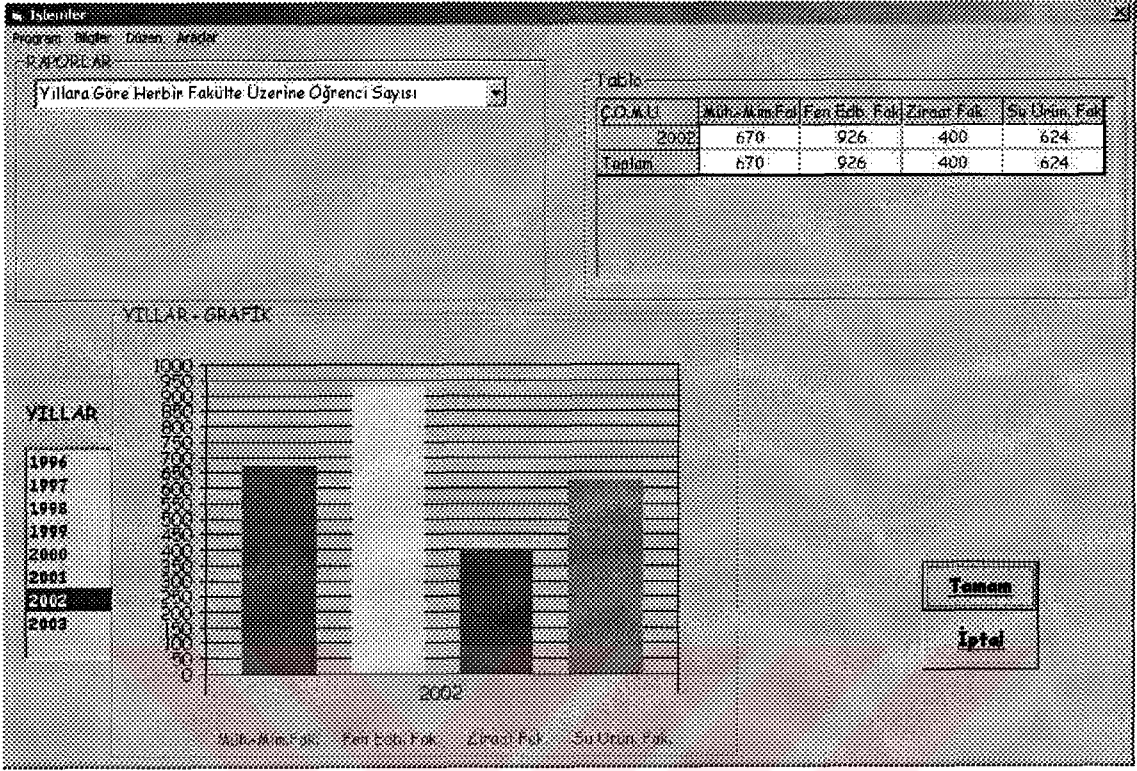
Birimler üzerinde makale sayısı toplamının bulunması:

Birimler bazında akademik personelin makale sayısı yıllar üzerinde hesaplanması işlemi gerçekleştirilir. Bu raporda birimler hiyerarşisine göre toplam makale sayısı hesaplanır. Bu rapor genişletilebilir. Biz sadece toplam sayıyı hesapladık. Fakat bu sayıyı makale türlerine ait bir sınıf yapısı üzerinde topladığımız zaman makale yayınlama türüne göre veya puanlamasına göre tekrar bir rapor oluşturulabilir.



Şekil 4.11 Makale Sayısı Liste Durumu

Bu rapor yönetici seviyesindeki kişilerin birimler hiyerarşisinde yukarıdan aşağıya doğru inildiğinde hangi yılda, hangi fakülteler ve hangi bölümler daha başarılı, daha aktif olduklarını öğrenebiliriz. Eğer bir düşme ile karşılaşılırsa sebep sonuç ilişkisine bakılabilir. Yönetici bu hesaplamaların ne şekilde gerçekleştiğini bilmemektedir. Yöneticiler sonuçların üzerinde değerlendirme yaparak karar verme aşamasında görev almaktadırlar.



Şekil 4.12 Yıllara göre herbir fakültede öğrenci sayısı

#### Yıllar Üzerine Ayrı Ayrı Fakülte Üzerine Öğrenci Sayısı:

Öğrenci sayısına göre şekil 4.8'deki arayüzde bir hesaplama yapılmıştır. Birimler hiyerarşisinde en üst seviyedeki üniversitede bulunmaktadır. Bir alt hiyerarşide fakülteler bulunmaktadır. Şekil 4.12'deki arayüzde üniversitedeki bütün fakülteler (bizim veritabanımızda mühendislik mimarlık, fen edebiyat, ziraat ve su ürünleri mevcuttur.)'in öğrenci sayılarını göstermektedir. Bu fakültelerin üzerinde tıkladığında ona ait bölümlerdeki öğrenci sayılarına ait grafik oluşturulur. Örneğin, Müh.-Mim.Fak. üzerinde tıklatıldığında, bilgisayar mühendisliği, gıda mühendisliği, jeoloji mühendisliği ve jeofizik mühendisliğine ait öğrencilerin sayısını gösteren grafik ile karşılaşırız.

Örnek uygulamada oluşturduğumuz raporları yukarıda arayüzleri ile birlikte açıkladık. Şimdi program üzerinde yapabileceğimiz değişiklikler veya yeni eklemeleri açıklayalım.

Menü Program, Bilgiler, Düzen, Araçlar kısmından oluşmaktadır. Program kısmında programdan çıkış ve kaydetme ile ilgili alanlar bulunmaktadır.

Bilgiler kısmında. Öğrenci Bilgileri, Dersler Hakkında Bilgiler, Yayınlar Hakkında Bilgiler, Akademisyenler Hakkında Bilgiler alanlarından oluşur. Öğrenci bilgilerinde öğrenci numarasına göre, dersler hakkında bilgiler ders koduna göre, yayınlar hakkında bilgiler yayinkodu ve alankoduna göre, akademisyenler hakkındaki bilgiler sicil nosuna göre sorgular yapılmaktadır.

Düzen kısmında üç boyutlu gösterme, etiket gösterme ve açıklama kısımlarından oluşur.

Araçlar kısmı yeni tablo ekleme ve yeni rapor ekleme kısımlarından oluşur.

Yeni tablo oluşturulması gerektiğinde izlenilmesi gereken aşamaları aşağıdaki gibidir:

Programın çalışması süresinde üzerinde çalışmalar yaptığımız veritabanına form üzerinden yeni tablonun oluşturulması gerekmektedir.

Visual Basic programlama dili kullanılarak aşağıdaki örnek oluşturulmaktadır.

```
Set db1 = Workspaces(0).OpenDatabase(App.Path + "\personel")
Set tablo = db1.CreateTableDef("yenitablo")
With tablo
    .Fields.Append .CreateField("Name", dbText)
    .Fields.Append .CreateField("Date", dbDate)
    .Fields.Append .CreateField("Phone", dbText)
End With
db1.TableDefs.Append tablo
Set recNew = db1.OpenRecordset("yenitablo")
With recNew
    .AddNew
    .Update
    .Close
End With
db1.Close
```

İşlemler

Program Bilgiler Düzen Araçlar

Yeni Tablo Oluşturma

Tablo İsmi : Yayımlar

Alan1 İsmi : YeniYayımlar Alan1 Tipi : Boolean

Alan2 İsmi: Alan2 Tipi : Integer

Alan3 İsmi: Alan3 Tipi : Currency

Alan4 İsmi: Alan4 Tipi : Single

Alan5 İsmi: Alan5 Tipi : Double

Text

Date

Oluştur

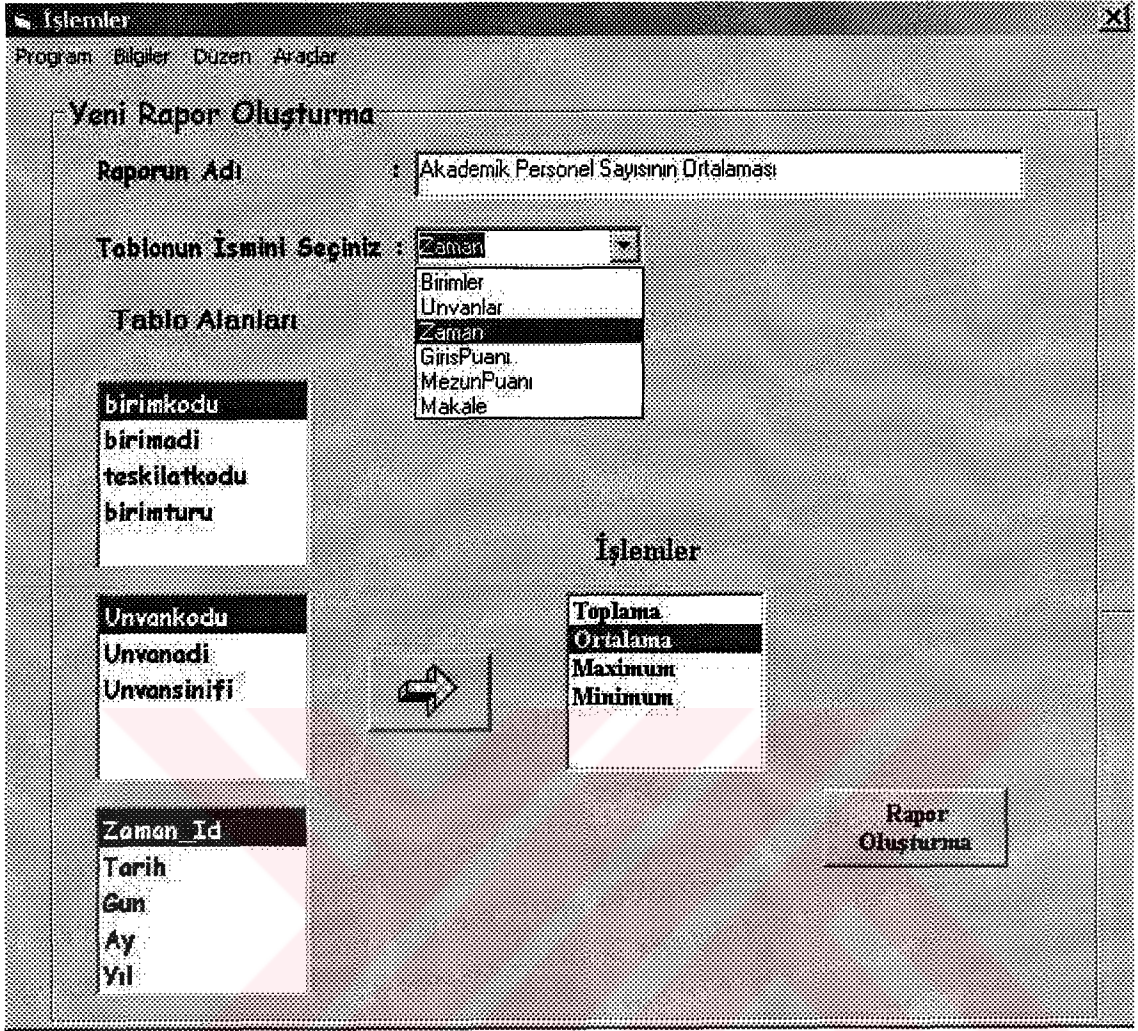
Şekil 4.13 Yeni tablo Oluşturma

Veritabanına programın dışından veritabanı açarak istenilen tablo kolaylıkla eklenmektedir. Fakat birden fazla işlem gerekmektedir. Bu işlemi form üzerinde tasarlayarak daha kolay işlem yapmamızı sağlamış olduk.

Tablonun ismini, alanlarını ve bu alanların değerlerini istenilen şekilde oluşturabiliriz. Tablo ekleme işlemi gibi yeni rapor ekleme işlemine gereksinim duyulur.

Hazır olarak forma eklediğimiz raporlar yeterli gelmeyebilir. Bunun için yeni bir form oluşturarak üzerinde gerekli işlemleri yapabiliriz. Bu işlem daha zordur. Çünkü oluşturulan raporda hangi tabloların kullanılacağı daha önceden belirlenmelidir. Yapılan işlemlerin daha iyi anlaşılması için ekran çıktısı olarak şekil 4.14'deki şekli inceleyelim.





Şekil 4.14 Yeni Rapor Oluşturma

Başlangıçta raporun adı girilir. Raporun oluşacağı tabloların isimleri seçilir. Her seçilen tablo alanları formda gösterilir. Tablo alanları seçildikten sonra bu alanlar istenilen alanlara sahip olduklarının SQL sorgularıyla doğrulandıktan sonra istenilen işlem uygulanarak `ambar_olusturma()` ve `grafik_ciz()` fonksiyonları işletilerek istenilen sonuca ulaşılır.

Bu uygulama geleceğe yönelik olarak genişletilebilir. Fakat kullandığımız veritabanı Access veritabanıdır. Bu veritabanının diğer veritabanları olan SQL Server, Oracle gibi firmalarla kıyaslandığından çok az sayıda veriyi depolayabilmektedir.

Görsel dil olarak visual basic kullanılmaktadır. Veri ambarı ve veri madenciliğine ait bileşenleri olmadığından bu işlemleri alt fonksiyonlar kullanarak gerçekleştirilmiştir.

SQL Server 2000 programlama ile veri ambarı oluřturma ve veri madencilięi iřlemleri yapmaya imkan saęlamaktadır. Hazır bileřenler üzerinde istenilen tablolar ve tablolara ait alanların seęimi ile bir sonraki adıma geęerek bylelikle istenilen yıldız Őema veya veri kpleri oluřturulur.

Yukarıda veri madencilięinde anlatılan SQL Server 2000 ile veri madencilięinde kullanılan iki tane yntem vardır. Veri kmeleme ve karar aęaęlarıdır. Bu ařamada elde edilen verilerin boyutları gz nnde bulundurularak bu iki yntem performans aęısından karřılařtırmakta mmkndr.

SQL Server veritabanında bulunan verilerle birlikte verilerin bir bięimden dięerine Microsoft Access yada Oracle gibi veritabanlarını tařınmasını saęlayan Enterprise Manager tarafından veri dnřtrme hizmetleri(DTS), veri almamızı ve vermemizi saęlayan grafik araęları ve nesnelere kullanılır. Bu da farklı veritabanlarındaki verileri tek SQL Server veritabanı mimarisinde toplamayı saęlamaktadır.

## SONUÇ

Bilgisayar teknolojisinin hızlı geliřimi ile bilgi kaynaklarının sayısı ok artmıřtır. Birok uygulamalarda karar destek tr uygulamaların gerekleřtirilmesinde zorluklarla karřılařılmıřtır. zellikle verilerin daęıtık Őekillerde farklı zelliklere sahip olması gçlkleri daha zorlařtırmaktadır. Bu yetersizlięin giderilmesi iin 90'lı yılların bařında, ilk olarak W.H. Inmon tarafından ortaya ıkarılan Veri Ambarı mimarisi oluřturulmaktadır. Veri ambarında veri ayrıntılardan arınıp zetlenerek veri analizini kolaylařtıracak Őekilde tutulur. Bu iřlemler gerekleřirken veri kaynaklarının tahmini, verilerin ıkarımında doęru verilere ulařmak, son kullanıcı gereksinimlerinin karřılanması gibi bazı zorluklar ve eksikliklerle karřılařılmaktadır.

Tezde aıklanan dięer konu veri madencilięidir. VT'ları veya VA'ları iinde bazı gizli ve doęrudan eriřilemeyen veriler bulunmaktadır. Verilerden faydalı bilgiler elde etmek iin geleneksel SQL sorgulama dili veya raporlama araęlarının son yıllarda yetersiz kalması Veri Madencilięi(VM) yntemlerinin geliřmesine neden olmuřtur

Tezin son blmnde Personel Bilgi Sisteminde veri ambarının gerekleřtirilmesi yapılmıřtır. Bu yazılım, Veri ambarı modelleri yıldız Őema ve veri kplerinin oluřturulması saęlamaktadır. Bu iřlemler ile verinin temizlenmesi ve

özetlenmesi işlemleri visal basic görsel dili kullanılarak raporda istenilen özetleme işlemlerinin gerçekleştirimi sağlanmaktadır.

Sonuç olarak; VA modelleri geleneksel VT modellerinden daha başarılıdır. VA yönetici seviyesinde kullanıcıya veri analizlerini sunduğunda geçen yıllar üzerindeki değişimleri görmek istemektedir. Ve bunun sonucunda geleceğe yönelik kararlar vermesinde yardımcı olmuştur.

Yapılan uygulama kısmında üst düzey kullanıcı olarak Akademik personel sayısındaki değişimlerin hesaplanması, öğrenci artışının hangi birimlerde daha fazla olduğunun, hangi birimlerde bilimsel faaliyetlerin daha fazla olduğunun hesaplanması yapılmıştır. Bu sonuçlar doğrultusunda gerekli analizlerin ve işlemlerin yapılması daha kolaylaşmış ve hızlanmıştır.



## ÖZET

Bu çalışmada, Veri Ambarı ve Veri Madenciliği teknolojileri ve mimarisi incelenerek bir uygulama alanında geliştirilerek hangi problemlere hangi çözümlerin sunulduğunun analizi ve uygulaması yapılmıştır.

İlk bölümde VTYS'lerinin uygulama özellikleri ve aralarındaki üstünlük ve eksiklikler araştırılarak kullanılan sistemler arasındaki farklılıklar incelenmiştir. Geleneksel veritabanlarında kullanılan Çevrimiçi İşlem İşlem(OLTP)'lerinde veritabanlarında ekleme, silme, güncelleme gibi işlemlerin gerçekleştiği uygulamalar dışındaki isteklere cevap verememesi nedeniyle daha ayrıntılı isteklere karşılık verebilecek yeni VTYS teknolojileri geliştirilmiştir. Bu teknolojilerin uygulama alanları veri özetleme, genelleme işlemleri ve ayrıca karmaşık sorguların gerçekleştirimine olanak sağlamıştır. Ardından gelen bölümde ise; yeni uygulamalarda bu işlemlerin gerçekleştirilmesiyle karar destek için analiz edilmiş bilgi üzerinde anlık sorgulama işlemlerinin tasarlanması sağlanmıştır. OLAP uygulamalarından ilişkisel OLAP ve çokboyutlu OLAP olarak yıldız şema mimarisi ve veri küp mimarisi incelenmiştir. Yıldız şemasının oluşturduğu boyut tabloları ve olgu tablosu arasındaki ilişkilere göre hesaplama değerleri elde edilir. Çokboyutsal veri küpü, veriye birden fazla boyutta bakmamızı sağlayarak veri üzerinde analizlerin kolay ve anlaşılır bir şekilde yapılmasını sağlar.

Üçüncü bölümde yapılan çalışmalar sonucunda şunlar söylenebilir. Teknolojinin gelişimiyle birlikte çok büyük veri birikimi oluşmaktadır. Bazı bilgiler VT'ları veya VA'larında gizli ve doğrudan erişilemez durumdadır. Bu bilgi elde etmede geleneksel SQL sorgulama dili veya raporlama araçları yetersiz kalmaktadır. Bu yetersizliklerin giderilmesi için yeni teknolojiler oluşturulmuştur. Yine bu bölümde incelenen Veri Ambarları üzerinden veri çıkarımı işleminin bir adımı olan Veri Madenciliği teknolojileri incelenmiştir. Veri madenciliği yöntemlerine, sınıflama, kümeleme, birliktelik kuralları ve ardaşık zamanlı örüntüler ve regresyon analizi örnekler verilebilir.

Dördüncü bölümde uygulama alanı olarak "Personel Bilgi Sistemi" üzerinde veri ambarı oluşturma ve veri madenciliği işlemlerinin gerçekleştirimi gösterilmiştir. Programlama dili olarak visual basic görsel dilinde örnek bir veri ambarı mimarisi oluşturulmuştur. Veri ambarı oluştururken bilgi sistemine ait veritabanları

kullanılmıştır. Bu bilgilerden istenilen raporlar sonucunda veritabanından bilgi çıkarımın bir adımı olan veri madenciliği uygulaması gerçekleştirilmiştir.

Örnek uygulama disket şeklinde teze ilave edilmiştir. Disket içerisindeki exe uzantılı dosyanın çalıştırılması yeterlidir.



## SUMMARY

In this thesis, Data Warehouse and data Mining technologies and structure was searched and these was extended by the application area for which solutions match which problems.

The first part, the applications' features of Database Management Ssystem and the priorities and differences between these systems were examined. In Online Transaction Processing, used traditional databases can not be taken result for any process without adding, deleting, upgrading etc.

For this reason, new DBMS which are responsible had improved. The application areas of these technologies are summarization of datas, process' of generalization and complex queries. After this part, adhoc query process', which datas was analysed for decision support with process realization in new applications were obtained in OLAP application, with relational OLAP and multi-dimensional OLAP star scheme and data cube structure was studied.

According relation between fact table and dimension tables which are becomed by star schemes, the equation values are obtained. Multi-dimensional data cube obtained to realize analysis on datas more clearly and easily by managing more than one dimension.

In third part, after prepared studies, these phrases can be said; with developing technologies expand data collections are occurred. Some informations can not be reached on Databases and Data Warehouses directly and hidden.

SQL and reporting equipment are insufficient for obtaining these kinds of informations. Because of removing these insufficiencies new techlogies were become. Furthermore, in same part, by the searched data warehouses Data Mining Technologies, which are the steps of data discovering process, was examined. Clustering, grouping, rules of sequence and consecutive timer patterns and regression analysis can be given as examples about data mining methods.

In fourt part, application area example was presented. This application is explain that data warehouse formation and realization of data mining process' on "Personal Information System" Visual Basic was selected by programming Language to establish a sample data warehouse structure.

While establishing data warehouse, databases belong to the information system were used. From this information, after looking reports for, the data mining application which is a step of discovering information was formed.



## **KAYNAKLAR**

- YARIMAĞA Ü.**, 2000. Veri Tabanı Sistemleri, Akademi & Türkiye Bilişim Vakfı Ortak Yayını, 362s.
- CONNOLLY T. M., CAROLYN E. B.** 1997. Database Systems (second edition), Addison-Wesley, 913-969p.
- RAMAKRISHMAN R., GEHRKE R.**, 2000. Database Management Systems, 677-732p.
- THURASINGHAM B.**, 1999. Data Mining Technologies, Techniques, Tools and Trend, CRC Presss, 270 p.
- VARDE S.A.**, 1999. Data Warehousing And Data Extraction On The World Wide Web, In Proceedings of WebTech 1999, USA.
- FIRESTONE M.J.**, 1997. Evaluating OLAP Alternatives 6p.
- SILBERSCHARTZ A., KORTH H.F, SUDARSHAN S.**, 1997. Database System Concepts, 697-708p.
- ULEY C**, 2000. Designing The Star Schema Database, (<http://www.ciobriefings.com/whitepapers/StarSchema.asp>), DSStar Company.
- ESTER M., WITTMANN R.**, Incremental Generalization For Mining In A Data Warehousing Environment, 1998. (<http://citeseer.nj.nec.com/205642.html>), Int. Conf. on Extending Database Technology.
- SHAO S.**, 1998.. Multivariate and Multidimensional OLAP, Advances In Database Technology-EDBT,98. 120-135p.
- KAYGULU M.S.**, 1999. Supervised And Unsupervised Learning Techniques In Data Mining, (Master Tezi).
- Information Discovery, Inc.**, 1997. A Characterization of Data Mining Technologies and Processes, <http://www.datamining.com/dm-tech.htm>.
- META GROUP**, 1996. An Introduction to Data Mining <http://www3.primushost.com/~kht/text/dmwhite/dbwhite.htm>.
- NGUYEN B.T, TJOA M., WAGNER R.**, 2000. An Object Oriented Multidimensional Data Model For OLAP (<http://citeseer.nj.nec.com/nguyen00object.html>), Institute of Software Technology(E188).
- DONGHUA X.**, 1997. A Survey On A Data Mining System: DBMiner



- DAL B., 1999. Veri Ambarı Ortamında Karar Destek Sistemi Tasarımı, ( Doktora Tezi).
- FIRESTONE M.J, 1997. Data Mining And KDD : A Shifting Mosaic.
- HAN J., CHIANG Y.J., CHEE S., CHEN J., 1999. DBMiner: A System For Data Mining In Relation Database And Data Warehouses  
(<http://www.cs.ualberta.ca/~zaiane/postscript/cascon97.pdf>).
- FIRESTONE J.M., 1997. Data Warehouses and Data Marts: A Dynamic View.
- FRANCONI E., SATTler U., 1999. A Data Warehouse Conceptual data Model for Multidimensional Aggregation  
(<http://citeseer.nj.nec.com/franconi99data.html>) Proceedings of The International Workshop On Design And Management of Data Warehouses(DMDW'99)
- RIORDAN M.R., 2001. Adım Adım Microsoft SQL Server 2000 Programlama, 665s.
- PC World, 1999. 70-82
- KARAGÜLLE İ., PALA Z., 1999. Visual Basic6.0 Pro, 948s.
- YÜKSEK Y., TAŞKIN E., 2003. Veri Ambarları Üzerinde Üniversite Bilgi Sisteminin Oluşturulması, Pamukkale Bilgi Teknolojisi Sempozyumu.
- SALAHLI M.A., 2003. Multilevel Architecture for Multidimensional Database Mathematical & Computational 263.
- GRAY J., CHAUDHURI S., BOSWORTH A.,1997. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals
- BONTEMPO C., ZAGELow G., 1998 The IBM Data Warehouse Architecture
- FAYYAD U., PIATETSKY G., SMYTH P., 1996. Knowledge Discovery and Data Mining: Towards A Unifying Framework

## **INTERNET ADRESLERİ**

<http://www.dkms.com>

<http://www.dmreview.com/>

<http://www.datawarehouse.com/>

<http://www.kdnuggets.com/>

<http://www.dmg.org/>

<http://www.data-mine.com/>

<http://salford-systems.com/>

<http://www.dw-institute.com/>

<http://www.datawarehouse.com/>

<http://www.cs.toronto.edu/~mendel/dwbib.html>

<http://www.billinmon.com/>

<http://citeseer.nj.nec.com/context/103687/0>

<http://www.lsi.upc.es/~aabello/references.html>



## TEŐEKKÜR

Bu tez alıőmasında bilgi ve deneyimlerinden yararlandıđım danıőmanım Do Dr. Mehmet Ali SALAHLI 'ya, gerekli imkanları veren sevgili Yrd.Do.Dr Servet SENYÜCEL 'e içtenlikle teőekkür ederim.

Tez yazımındaki yardımları için arkadaşım Canay GÜMÜŐLÜ'ye içtenlikle teőekkür ederim.



## **ÖZGEÇMİŞ**

Adı Soyadı : Yasemin YÜKSEK

Doğum Yeri ve Yılı : Çanakkale 30.05.1978

Adres :ÇOMU Mühendislik-Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümü

### **Eğitim Durumu**

1984-1989:Çanakkale Gazi İlkokulu

1989-1992: Çanakkale Merkez OrtaOkulu

1992-1995: Çanakkale İbrahim Bodur Lisesi

1996-2000: ÇOMU Mühendislik-Mimarlık Fakültesi Bilgisayar Mühendisliği

2000-2003: ÇOMU Fen Bilimleri Enstitüsü Bilgisayar Yazılım Anabilim Dalı

### **Staj ve Kurslar**

01.07.1999 – 10.08.1999 Çanakkale Akçansa Çimento Fabrikası Bilgi İşlem Daire Başkanlığı

### **Mesleki Deneyim**

2000-... : ÇOMU Mühendislik-Mimarlık Fakültesi Bilgisayar Mühendisliği Araştırma Görevlisi

### **Çalışma ve İlgili Alanları**

Veritabanı Yönetim Teknolojileri

Veri Yapıları

Web Teknolojileri