

127023

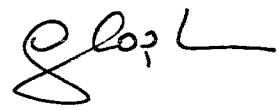
VERİTABANI TEKNOLOJİSİ İLE KENT BİLGİ
SİSTEMLERİ TASARIMI

YÜKSEK LİSANS TEZİ
Müh. Zehra Damla UÇA
(514981069)

TC. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

Tezin Enstitüye Verildiği Tarih : 2 Ocak 2002
Tezin Savunulduğu Tarih : 18 Ocak 2002

Tez Danışmanı : Doç.Dr. Gültekin ÇETİNER
Diğer Jüri Üyeleri : Doç.Dr. Şakir KOCABAŞ
Doç.Dr. Gonca COŞKUN


Ş. Kocabaş

Gonca Coşkun

127023

OCAK 2002

ÖNSÖZ

Kent Bilgi Sistemleri uygulamalarının hayata geçirilmesi ile, ilgili çok çeşitli kurumun, mevcut yöntemlerle karşılaştırılamayacak kadar fazla fayda sağlayacağı bir gerçektir. Bu sistemlerin internet teknolojileri ile tümleştirilmesi ise, erişimi kolaylaştıracak ve kullanıcı sayısını milyonların üstüne taşıyacaktır. Bu tezde yapılan çalışmanın, bu tür çalışmalar için bir kaynak olması planlanmıştır.

Tez çalışmam süresince bana yardımcı olan çok değerli hocam Sayın Doç. Dr. Gültekin ÇETİNER'e, başta Yrd. Doç. Dr. Temel Aygün, Doç. Dr. Şakir Kocabaş ve Doç. Dr. Necla Uluğtekin olmak üzere ders aşamamda emekleri geçen hocalarıma, değerli vakitlerini ayıran Dr. Levent Topaktaş'a ve bana her zaman destek olan aileme, Y. Müh. Murat Avcı'ya ve ilgilerini esirgemeyen tüm herkese sonsuz teşekkürler...

Ayrıca bana, tez uygulaması kapsamında kullanmak üzere Eskişehir iline ait verileri sağlayan Kadastro Müdürlüğü'ne, akademik çalışmalar için deneme versiyonları ile yazılım sağlayan Intergraph Firmasına, internet teknolojileri konusunda öncü kuruluş Parkyeri A.Ş.'ye, sahip olduğu ülkemizin en gelişmiş kent bilgi sistemlerinden biri ile bana örnek olan Eskişehir Belediyesi'ne, Anadolu Üniversitesi'nden Prof. Dr. Engin Ataç ve Prof Dr. Nezih Varcan'a ve örnek bir akademik kurum olduğuna inandığım İstanbul Teknik Üniversitesi'ne teşekkür ederim.

Bu tez, çalışkan ve dürüst bilim adamları ile, yeni teknolojiler üretme ve ülkesine kazandırma yolunda yorulanlara adanmıştır.

OCAK, 2002

Müh. Zehra Damla UÇA

İÇİNDEKİLER

KISALTMALAR	vii
TABLO LİSTESİ	viii
ŞEKİL LİSTESİ	ix
ÖZET	xii
SUMMARY	xiii

1. GİRİŞ	1
2. COĞRAFI BİLGİ SİSTEMLERİ	4
2.1 Haritalar	4
2.1.1 Harita Tanımı ve Sağladıkları	4
2.1.2 Haritanın Tarihsel Gelişimi	4
2.1.3 Harita Dizaynı	6
2.1.4 Harita Çeşitleri	7
2.1.4.1 Fonksiyonlarına Göre Harita Çeşitleri	7
2.1.4.2 Oluşturulma Metotlarına Göre Harita Çeşitleri	7
2.1.4.3 Konularına Göre Harita Çeşitleri	8
2.1.4.4 Ölçeklerine Göre Harita Çeşitleri	8
2.1.5 Haritada Konum Bilgisinin Sağlanması	9
2.1.5.1 Coğrafi Konumlandırma Metotları ve Karşılaştırılmaları	10
2.1.5.2 Coğrafi Koordinat Sistemleri, Enlemler, Boylamlar	11
2.1.5.3 Harita Projeksiyonları	11
2.2 Coğrafi Bilgi Sistemleri	15
2.2.1 Konumsal Verinin Dijitalleşmesi ile Haritanın Yeni Ortamı	15
2.2.2 Coğrafi Bilgi Sistemleri Bileşenleri	15
2.2.2.1 Coğrafi Veriler	16
2.2.2.2 CBS Donanım Bileşenleri	16
2.2.2.3 CBS Yazılım Bileşenleri	17
2.2.2.4 CBS Personeli	20
2.2.3 CBS Yapısı	21
2.2.3.1 CBS Projesi Aşamaları	21
2.2.3.2 Katman Mantığı	21
2.2.4 CBS Verisi Bileşenleri	23
2.2.4.1 CBS Verisi Geometrik Bileşeni	24
2.2.4.2 CBS Verisi Niteliksel Bileşeni	24
2.2.4.3 CBS Verisinin 2. Bileşeni Olarak Zaman	25
2.2.5 CBS Kullanımı	25
2.2.5.1 CBS Kullanımı Gerektiren 3 Durum	25
2.2.5.2 CBS Kullanımı ile Cevaplanabilen Sorular	25
2.2.5.3 CBS Kullanım Alanları	27
2.2.5.4 CBS Kullanımında Mevcut Sorunlar	29
2.2.5.5 Harita ve CBS'nin Geleceği	30
2.3 CBS ile Veriden Ürün Harita Elde Etme	31
2.3.1 CBS için Veri Elde Etme Yöntemleri	31
2.3.2 Hazırlanan Verilerin Birleştirilmesi	32
2.3.3 Verinin CBS Ortamına Aktarımı	33
2.3.4 Verinin Modellenmesi	34

2.3.4.1 Vektör Veri Yapısı Yaklaşımı	34
2.3.4.2 Raster Veri Yapısı Yaklaşımı	36
2.3.4.3 Hem Vektör Hem Raster Veri Yapısının Bir Arada	
Kullanılması	38
2.3.5 CBS Verisinin Görselleştirilmesi	40
2.3.6 Görselleştirilen Veri ile Çalışmak	44
2.3.7 CBS ürünü: Harita	45
2.3.7.1 Konumsal Verinin Geometrik Bileşenini Karşılaştırmak	45
2.3.7.2 Konumsal Verinin Niteliksel Bileşenini Karşılaştırmak	45
2.3.7.3 Konumsal Verinin Zamansal Bileşenini Karşılaştırmak	45
2.3.7.4 Ürün Haritada Sonucun Kullanıcıya Hissettirilmesi	46
3. VERİTABANI SİSTEMLERİ	48
3.1 Veritabanları	48
3.1.1 Veritabanı	48
3.1.2 Veritabanı Sistemlerinin Gelişimi	49
3.1.2.1 Klasik Dosya Sistemleri	49
3.1.2.2 Klasik Dosya Sistemlerinin Dezavantajları	49
3.1.2.3 Veritabanı Sistemlerinin Avantajları	51
3.1.3 Veritabanı Yönetim Sistemleri	52
3.1.3.1 Veritabanı Yönetim Sistemleri Bileşenleri	52
3.1.3.2 Veritabanı Yönetim Sistemleri Fonksiyonları	53
3.1.3.3 Veritabanı Yöneticileri	54
3.1.4 Veritabanı Sisteminin Çalışması	55
3.1.4.1 Veritabanı Kullanıcıları	56
3.1.4.2 Uygulama Programları	56
3.1.4.3 Meta-Data	56
3.1.4.4 Sorgu Prosesi	57
3.1.5 İlişkisel Veritabanı Yönetim Sistemleri	58
3.1.6 Veritabanı Dilleri	59
3.1.6.1 Veri Tanımlama Dili	60
3.1.6.2 Veri Yönetim Dili	60
3.2 Veritabanı Sistemlerine Soyut Yaklaşım	61
3.2.1 Veri Görünümü	61
3.2.2 Veri Soyutlama	61
3.2.3 Veri Modelleri	63
3.3 Sistem ve Veritabanı Geliştirme	68
3.3.1 Sistem Geliştirme	68
3.3.2 Veritabanı Geliştirme	69
3.4 Veri Modelleme	70
3.4.1 Veri Modeli Oluşturma	70
3.4.1.1 Varlıklar	70
3.4.1.2 Özellikler	72
3.4.1.3 Eşsiz Tanımlayıcı Atama	73
3.4.1.4 İlişkiler	74
3.4.2 Veri Modeli Atama	81
3.4.2.1 Veri Modelini Normalize Etme	81
3.4.2.2 Çoktan-çoğa İlişkileri Çözümleme	83
3.4.2.3 Tekrarlı İlişkileri Modelleme	85

3.4.3 İleri Seviyeli İlişkiler	86
3.4.3.1 Özel İlişkileri Modelleme	86
3.4.3.2 Alt Tipleri ve Süper Tipleri Tanımlama	88
3.4.4 Veri Modeli Dönüşümü	89
3.4.4.1 İlişkisel Veritabanı	89
3.4.4.2 Varlık-İlişki Diyagramını Fiziksel Veritabanına	
Dönüştürme	92
3.5 SQL (Structured Query Language)	93
3.5.1 SQL Dili	93
3.5.2 SQL Komutları	94
3.5.3 Embedded SQL	98
3.5.4 SQL Server	98
4. KENT BİLGİ SİSTEMLERİ	100
4.1 Kent Yönetiminde KBS İhtiyacı	100
4.1.1 Kent Yönetiminde İhtiyaçlar	100
4.1.2 Bilgi Sistemleri ve Coğrafi Veri	101
4.1.3 KBS Kullanımı	102
4.1.4 KBS Kullanımı ile İlişkili Temel Kuruluşlar	103
4.2 KBS Teknolojisinin Sağladıkları	105
4.2.1 Maliyet / Yarar Analizi	105
4.2.2 KBS Sistemlerinin Sağladığı Avantajlar	107
4.3 Mevcut KBS Örnekleri	108
4.4 Bir KBS Projesi Oluşturma	109
4.4.1 Kullanılan Bilginin Yapısı	109
4.4.2 Amaç Belirleme	111
4.4.3 Kent Öğelerini ve Özelliklerini Belirleme	112
4.4.4 Ayrı Katmanlara Veri Girişi	113
4.4.5 Sistemin Veritabanı Tarafı	116
4.4.6 Sorgulama	118
4.5 KBS ile Tümüleştirilmiş Sistemler	123
5. KENT BİLGİ SİSTEMLERİNİN İNTERNET TEKNOLOJİLERİ İLE TÜMLEŞTİRİLMESİ	125
5.1 İnternet Teknolojileri	125
5.1.1 İnternetin Tanımı ve Tarihçesi	125
5.1.2 İnternet Kullanımı	126
5.1.3 FTP Aracı, WWW Standardı ve Web Sayfaları	127
5.1.4 HTTP Protokolü ve URL Adresi	127
5.1.5 Web Tarayıcısı	128
5.1.6 Web Sunucusu ve İnternet Servis Sağlayıcılar	128
5.1.7 HTML Dosyaları, HTML Dili ve Kodlar	129
5.1.8 JavaScript, VBScript, ASP Dili ve ActiveX Teknolojisi	130
5.1.9 IIS Yazılımı ve Yönetimi	132
5.2 KBS'nin Web Sunucu Sistemine Kurulması	132
5.2.1 KBS Verisi	132
5.2.2 GeoMedia Web Map Yazılımı, ACGM Kontrolü	133
5.2.2.1 Web-Sunucu Donanımı	133
5.2.2.2 Web-Sunucu Yazılımları	134

5.2.2.3 ActiveCGM Kontrolü	134
5.2.3 Sistemin Kurulumu	135
5.2.3.1 IIS Yüklenmesi ve Ayarlar	135
5.2.3.2 GMWM Yüklenmesi ve Kurulumu	136
5.2.3.3 Verinin Hazırlanması	136
5.2.3.4 Web Sayfası Tasarımı	138
5.2.4 Sistemin Çalışması	139
6. SONUÇLAR VE TARTIŞMA	157
KAYNAKLAR	159
ÖZGEÇMİŞ	161



KISALTMALAR

CBS	: Coğrafi Bilgi Sistemleri
KBS	: Kent Bilgi Sistemi
CAD	: Computer Aided Design
GPS	: Global Positioning System
ANSI	: American Standarts National Institue
ISO	: International Standarts Organization
SQL	: Structurel Query Language
DBMS	: Database Management System
VTYS	: Veritabanı Yönetim Sistemi
DBA	: Database Administrator
DDL	: Data Definition Language
DML	: Data Manipulation Language
RDBMS	: Relational Database Management System
E-R	: Entity-Relationship
UA	: Uzaktan Algılama
AI	: Artificial Intelligence
TCP/IP	: Transmission Control Protocol / Internet Protocol
ARPA	: Advanced Research Projects Agency
ARPANET	: Advanced Research Projects Agency Network
FTP	: File Transfer Protocol
PC	: Personel Computer
WWW	: World Wide Web
CERN	: European Laboratory For Practice Physics
HTTP	: Hypertext Transfer Protocol
HTML	: Hypertext Markup Language
URL	: Uniform Resource Locators
ISP	: Internet Service Provider
IIS	: Internet Information Services
ASCII	: American National Standard Code For Information Interchange
CGI	: Common Gateway Interface
ASP	: Active Server Pages
JPEG	: Joint Photographic Experts Group
CGM	: Computer Graphics Metafile
ACGM	: Active Computer Graphics Metafile
IE	: Internet Explorer
NN	: Netscape Navigator
CSF	: Coordinate System File
MDF	: Map Definition File

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 3.1 Veritabanında oluşturulmuş ‘çalışanlar’ tablosu	57
Tablo 3.2 Sorgu sonucunu gösteren liste	58
Tablo 3.3 ‘Müşteriler’ tablosu	65
Tablo 3.4 ‘Hesaplar’ tablosu	65
Tablo 3.5 Örnek bir veritabanı tablosu	90
Tablo 3.6 Tablo ve birincil anahtar olarak kullanılacak sütun	90
Tablo 3.7 ‘Fakülte’ tablosunda birincil anahtar	91
Tablo 3.8 ‘Sınıf’ tablosunda yabancı anahtar kullanılması	91
Tablo 3.9 Varlık- ilişki modelindeki kavram ve nesnelerin ilişkisel veritabanına dönüşümü	92
Tablo 3.10 ‘Bölüm’ tablosu	93
Tablo 3.11 ‘Personel’ tablosu	93

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1	Oluşturulma metotları farklı çeşitli haritalar görülmektedir. 8
Şekil 2.2	Coğrafi konumlandırma metotları, avantaj ve dezavantajları görülmektedir. 10
Şekil 2.3	Azimutal, silindirik ve konik projeksiyonlar ile bunların normal, dik ve eğik yerleştirilmeleri 12
Şekil 2.4	Grönland ve Suudi Arabistan'ın alan koruyan ve şekil koruyan projeksiyonlarla gösterimi. 14
Şekil 2.5	Geometrik transformasyon işlemleri görülmektedir..... 14
Şekil 2.6	Coğrafi Bilgi Sistemlerinde katman mantığı 22
Şekil 2.7	CBS sistemlerinde katman-tablo ilişkisi 23
Şekil 2.8	Gerçek dünyadaki coğrafi nesnelerin CBS ortamına aktarımı 24
Şekil 2.9	Bir çizginin vektör ve raster formatlarda gösterimi 33
Şekil 2.10	Vektör veri yapısı ve veritabanındaki ifadesi 36
Şekil 2.11	Raster veri yapısını oluşturan hücreler 38
Şekil 2.12	Farklı veri yapılarının sorgulamalarda sağlayacağı esneklikler 39
Şekil 2.13	Coğrafi nesnelerin noktasal, çizgisel, alansal ve hacimsel gösterimi 41
Şekil 2.14	Coğrafi nesnelerin 2, 3 ve 4 boyutlu gösterimleri 42
Şekil 2.15	Sınıflandırılabilir ve sürekli veri gösterimi 43
Şekil 2.16	Yağış miktarları dağılımının iki farklı yöntemle gösterimi 46
Şekil 3.1	Veritabanı kullanımı sırasında gerçekleşen aşamalar 55
Şekil 3.2	Görünüm seviyesi, mantıksal seviye ve fiziksel seviye arasındaki ilişkiler 62
Şekil 3.3	E-R diyagramından örnek bir varlık-ilişki yapısı 64
Şekil 3.4	Hiyerarşik veri modeli yapısı 66
Şekil 3.5	Ağ veri modeli yapısı 67
Şekil 3.6	Sistem geliştirme aşamaları 68
Şekil 3.7	Veritabanı geliştirme aşamaları 69
Şekil 3.8	Varlıkların E-R diyagramında gösterimi 71
Şekil 3.9	Bir varlığa ait özelliklerin diyagramda gösterimi 73
Şekil 3.10	Bir varlığa ait eşsiz tanımlayıcının E-R diyagramında gösterimi ... 73
Şekil 3.11	Bir varlık için iki eşsiz tanımlayıcının gösterimi 74
Şekil 3.12	Varlıklar arası ilişkilerin gösterimi 74
Şekil 3.13	İlişki cümleciklerinin yapısı 75
Şekil 3.14	İlişkilerin genel gösterimi 76
Şekil 3.15	Zorunlu ve seçime bağlı ilişkilerin gösterimi 76
Şekil 3.16	İlişki derecelerinin gösterimi 76
Şekil 3.17	Birden-bire bir ilişkinin yapısı 77
Şekil 3.18	Birden-bire bir ilişkinin varlık-ilişki diyagramında gösterimi 78
Şekil 3.19	Birden-çoğa veya çoktan-bire ilişkilerin yapısı 78

Şekil 3.20	Birden-çoğa veya çoktan-bire ilişkilerin varlık-ilişki diyagramında gösterimi	79
Şekil 3.21	Çoktan-çoğa ilişkilerin yapısı	79
Şekil 3.22	Çoktan-çoğa ilişkilerin varlık-ilişki diyagramında gösterimi	79
Şekil 3.23	Tanımlayıcı bir ilişkinin varlık-ilişki diyagramında gösterimi	80
Şekil 3.24	Birinci normal forma uyan bir varlık diyagramı	81
Şekil 3.25	İkinci normal forma uyan bir varlık diyagramı	82
Şekil 3.26	Üçüncü normal forma aykırı varlıklar	82
Şekil 3.27	Çoktan-çoğa bir ilişkinin varlık-ilişki diyagramında gösterimi	83
Şekil 3.28	Çoktan çoğa bir ilişkiye eklenen kesişim varlığı	84
Şekil 3.29	Bir kesişim varlığı ile çözümlenmiş çoktan çoğa bir ilişkinin varlık-ilişki diyagramında gösterimi	85
Şekil 3.30	Tekrarlı bir ilişkiye ait varlık-ilişki diyagramı	86
Şekil 3.31	Tekrarlı ve çoktan çoğa bir ilişkinin varlık-ilişki diyagramında gösterimi	86
Şekil 3.32	Özel ilişkilerin varlık-ilişki diyagramında gösterimi	87
Şekil 3.33	Alt tip ve süper tiplerin varlık-ilişki diyagramında gösterimi	88
Şekil 3.34	Süper ve alt tiplerin ayrı ayrı başka varlıklar ile ilişkileri olması durumu	89
Şekil 3.35	Örnek bir E-R diyagramı	92
Şekil 4.1	Kent bilgi sistemleri kullanımında ilişkili olan kurum ve kuruluşlar	105
Şekil 4.2	Kurum ve kuruluşların yaşam sürecinde görülen KBS yararları	106
Şekil 4.3	Zaman içinde KBS ve klasik yönetim sistemlerinin kümülatif maliyetleri	107
Şekil 4.4	Örnek bir KBS’de coğrafi verinin görünümü	111
Şekil 4.5	Veri kümeleri ve içerdikleri örneklerin katman yapısında saklanması	113
Şekil 4.6	Katmanların üst üste açılması ve altlık harita kullanılması	114
Şekil 4.7	Veritabanı tarafındaki katman yapısının sembolize edilmesi	115
Şekil 4.8	Veritabanında veri kümesi örneklerinin satır, özelliklerinin ise sütunları oluşturması	116
Şekil 4.9	KBS, hem grafik veriler hem de bunlarla ilişkili veritabanının bütünüdür	117
Şekil 4.10	Örnek bir kampus bilgi sistemi için grafik elemanlar ve veritabanı ilişkisi	118
Şekil 4.11	Bir sorgulama sürecinde veritabanının taranarak sonuç listesinin çıkarılması ve yeni harita üretilmesi	120
Şekil 4.12	Veritabanındaki değerlerin sınıflandırılarak gösterimi	121
Şekil 4.13	Bir veri grubuna belli uzaklıktaki çevre bölgenin gösterimi	122
Şekil 4.14	Harita penceresi ile veri penceresinin aynı anda izlenebilmesi	123
Şekil 4.15	Eskişehir kent bilgi sisteminden bir görüntü	124
Şekil 5.1	Eskişehir Kent Bilgi Sisteminde genel Eskişehir görüntüsü	140
Şekil 5.2	Tepebaşı Belediyesi Mahalleleri	141
Şekil 5.3	Odunpazarı Belediyesi Mahalleleri	142
Şekil 5.4	Hoşnudiye Mahallesi’ne tıklayarak ulaşılan ilgili bir web sayfası ..	143
Şekil 5.5	Tepebaşı Belediyesi mahalleleri ve kent öğeleri	144
Şekil 5.6	ACGM kontrolleri ve bunları kullanarak yaklaştırılan harita	145
Şekil 5.7	Ekrandaki harita ve magnify penceresinin bir arada kullanılması ..	146

Şekil 5.8	Akarbaşı Mahallesi	147
Şekil 5.9	Harita üzerinde işaretleme yapmak üzere kullanılan menü	147
Şekil 5.10	Ziyaretçinin harita üzerinde çizim ve yazım ile işaretleme yapması	148
Şekil 5.11	Deliklitaş Mahallesi civarındaki öğrenci yurtları	149
Şekil 5.12	Sorgu sonucu erkek öğrenci yurtlarının gösterilmesi	149



ÖZET

Coğrafi Bilgi Sistemleri uygulamaları, coğrafi verileri konumsal ve/veya öznitelik bilgileri yönünden değerlendirerek bilgi ve harita üretmenin en kolay, hızlı ve güvenli yoludur. Kent düzeyinde bilgi içeren özel çalışmalar ise, “Kent Bilgi Sistemleri” olarak bilinir. Kent Bilgi Sistemleri çok çeşitli analizler ve farklı amaçlar için kullanılır.

Pek çok dalda araç olarak kullanılmasının yanı sıra kent ile ilgili genel bir bilgi sistemi olarak da tasarlanmaktadır. Kent Bilgi Sistemlerinin internet teknolojileri ile tümleştirilmesi, bu hizmetin çok sayıda kullanıcının paylaşımına en yaygın bilgisayar ağı ile açılmasını sağlayacaktır.

Bu tezde, kent bilgi sistemlerine duyulan ihtiyaç ve kent bilgi sistemlerinin yapısı anlatılmıştır. Bu sistemler, arka planda veritabanında tutulan bilgilere bağlı elemanlardan oluşan ve katman yapısı ile saklanan vektörel haritaların üretildiği ve sorgulanabildiği sistemlerdir.

Tezin uygulama kısmında, bir Kent Bilgi Sisteminin internet üzerinden yayınlanması uygulaması yer almaktadır. Bir kent bilgi sistemini internet üzerinden sunmak, sunucu bir işletim sistemi aracılığı ile gerçekleştirilir. Web üzerinden bağlanan kullanıcıya sorgulama imkanı tanıyan bu sistemde, kullanıcının isteği bir kodlama dili aracılığı ile sisteme aktarılmakta ve veritabanına ulaşılarak çıkarılan sorgu sonuçlarına göre yeniden harita üretilerek istemci makineye gönderilmektedir.

SUMMARY

Using Geographical Information Systems (GIS) technology is the most fast, easy and reliable way of evaluating the spatial component and attributes of geographical data. If the information system is structured to involve information of a specific city, it is called an “Urban Information System”. Urban Information Systems are used practically for various purposes and analysis by relevant associations and foundations.

One of the many ways of using urban information systems is to design the system to serve general information to many users as can be. These systems will contain not detailed but basic information related with the main geographical elements on the digital map. Integration of GIS and internet technologies provide an opportunity to serve the system to users on an international wide network.

In this study, the need of urban information technologies on today’s world is mentioned. The thesis informs about the structure of the urban information systems.

On the application part, a web page is designed that serves a sample urban information system from a web page. To serve GIS on internet requires to be on a server operating system. This system provides the client a query chance. The client’s request, the query question is transferred to the system by a programming language, then reached to the database, the results of the query is obtained and the requested information is produced as a new map displaying results and sent to the client side.

1. GİRİŞ

Coğrafi Bilgi Sistemleri (CBS), coğrafi verinin harita üzerindeki grafik bileşeni ile veritabanındaki nitelik bileşenini bir arada kullanma, sorgulama ve sunma teknolojisidir. CBS uygulamalarının bilgisayar ortamında gerçekleştirilmesi, gün geçtikçe, coğrafi veri kullanan tüm disiplinler için CBS kullanımını kaçınılmaz kılmaktadır.

CBS projeleri yeryüzünü veri kaynağı olarak kullanan modellerdir. Kullanılan veriler, uzayda belirli bir konuma sahip olan verilerdir. CBS ortamında, verinin geometrik bileşeni ve niteliksel bileşeni vardır. Geometrik bileşen verinin grafik görüntüsü, koordinatları, geometri bilgisidir ve harita olarak ortaya çıkar, niteliksel bileşen ise, öznitelik bilgileridir ve veritabanında saklanır. Bu sebeple, coğrafi bilgi sistemleri teknolojisi, veritabanı teknolojisi ile çok ilgilidir.

CBS 'nin en belirgin ve etkili özelliği katman yapısıdır. CBS 'nin en çok kullanıldığı durum iki veri grubu arasında ne gibi bir ilişki olduğunu ortaya çıkarmaktır. CBS 'de ilişkilendirmeler ortaya çıkarılırken katman yapısı çok kullanışlı olmakta, analiz için gerekli katmanlarla çalışma imkanı sağlamaktadır.

Yönetimlerde kullanılan karar destek sistemi içerisinde bilgisayar teknolojisi yoksa, istihdam fazlalığı ve gereksiz bazı işlemlerden kaynaklanan zaman kaybı ve parasal kayıplar olacaktır. Veri toplama, birleştirme, güncelleştirme, haritaya işleme, arşivleme, bilgi türetme gibi verilerle ilgili tüm işlemlerde CBS, en verimli yöntemdir.

Pek çok konuda bir araç olarak kullanılan CBS, hızlı öngörü sağlamada mükemmel bir araçtır. Coğrafi veri tanımı ne kadar geniş ise, CBS uygulama alanlarının da o denli çok olacağı açıktır.

Gelişmiş ülkelerde, CBS sistemlerinin tüm ülke için hazırlanması ulaşılmak istenen hedeftir. Ancak, hem ülkemizde hem de dünyada böyle sistemler kurulmadan önce, kent bilgi sistemleri kurulmalı ve bu teknolojiye hiç olmazsa şehir bazında yaşanan problemlerin çözümünde yararlanılmalıdır.

Kent yönetiminde etkin olabilmek için kullanılan yöntemler sağlıklı, etkin, hızlı, ucuz ve verimli olmalıdır, kent bilgi sistemleri CBS teknolojisinin kent düzeyinde özelleşmiş uygulamasıdır ve bu ihtiyaçlara cevap vermektedir.

Kent Bilgi Sistemleri (KBS) pek çok kurum ve kuruluşu ilgilendirmektedir. Kent Bilgi Sistemini kullanacak ve ilişkiler ağı içerisinde yer alacak temel kuruluşlar valilikler, belediyeler, tapu ve kadastro müdürlükleri, su ve kanalizasyon idaresi, gaz idaresi, toplu taşımacılık idaresi, elektrik idaresi, telefon idaresi olacaktır. KBS sisteminin oturtulması ile zaman, para ve insan gücü tasarrufu sağlanacaktır.

KBS sistemleri kurumların analiz yapmaları, bilgilerini ortaya koymaları ya da kararlara varmaları için kullanılır. Bunun yanı sıra, Kent Bilgi Sistemlerinin kullanım amaçlarından biri de başta kentlilere olmak üzere, kent ile ilgili herkesin yararlanabileceği bir bilgi sistemi sağlamak olabilir. Böyle bir bilgi sisteminde, bir belediye bilgi sisteminde olduğu detayda bilgi bulunmayacaktır. Bu sistemler temel kent öğeleri hakkında, genel bilgiler içereceklerdir. Diğer yandan bu tür bilgi sistemlerinin amacı genel bir hizmet olduğundan, mümkün olan en geniş kitleye, en hızlı ve kolay erişimi sağlayan yaygın bir sistem ile sunulmalıdır. Herkesin erişimine açık ve kolay ulaşılabilir olması, interneti genel hizmet amaçlı kent bilgi sistemlerini sunmanın en avantajlı yöntemi haline getirmiştir.

Bugün büyük bir hızla gelişmekte olan internet ve web teknolojileri, web üzerinden veritabanı ilişkili verilerin sunulmasına imkan vermektedir. KBS 'nin internet teknolojisi ile tümleştirildiği uygulamalarda, KBS sistemi, bir sunucu sisteme kurulur. Kentteki coğrafi verilerin geometrik bileşenlerinden üretilen ve ekranda görüntülenen harita, arka planda her elemanın öznitelik bilgilerinin tutulduğu bir veritabanına bağlıdır. Sistem, kullanıcılara sağlanan esneklik ölçüsünde sorgulamaya müsaittir. Gelen isteğe bağlı olarak arzulanan harita üretilir ve istemci makineye gönderilir.

Internet üzerinden sunulacak bir KBS 'nin tasarımında, öncelikli amaç ve hizmet olarak verilmek istenen nedir, bu sistemin kullanımı ile sağlanacak kolaylıklar

nelerdir, sistemin gvenlięi nasıl saęlanmalıdır, kullanıcıya grsel ynden hitap edecek ve kullanımda da pratik olacak sayfa tasarımı nasıl olmalıdır, arka planda olmasına raęmen sistemin altyapısını, temelini oluřturan veritabanı ne řekilde zellikler ieren, hangi zelliklerde bir veritabanı olmalıdır gibi soruların cevapları aranarak bir kent bilgi sistemi tasarlanmalıdır.

Kent Bilgi Sistemlerinin sadece analiz ve planlama yapan kuruluřlar iin deęil, herkes iin kent hakkında bilgi edinmenin bir aracı olarak kullanılması ve bu tr sistemlerin arttırılması byk faydalar saęlayacaktır.



2. COĞRAFI BİLGİ SİSTEMLERİ

2.1 HARİTALAR

2.1.1 Harita Tanımı ve Sağladıkları

Haritalar, konumsal verinin görselleştirilmesinde kullanılan araçlardır. Son yıllarda bu tanımın kapsamı çok genişlemekle beraber, genel olarak yeryüzünün belirli bir bölümünün ya da tümünün coğrafi, jeolojik, ya da jeopolitik özelliklerinin belirli ölçekler içinde düz bir yüzey üstüne çizilmesi yoluyla elde edilen grafik gösterim olarak tanımlanabilirler. [1]

İnsanlara harita kullanma gereksinimi hissettiren bazı ihtiyaçları olmuştur. Haritalardan en temel anlamda yön, uzaklık ve alan bilgileri elde edilebilir. Haritalar, konumsal ilişkileri daha iyi anlama fırsatı sağlar.

2.1.2 Haritanın Tarihsel Gelişimi

Tarihte harita kullanımı incelendiğinde, iletişim aracı olarak çok eskilerden beri kullanıldığı göze çarpmaktadır. “İnsanlığın ortaya koyduğu 3 iletişim aracı dil, müzik ve haritadan en eski olanı haritadır.” [2]

Bilinen ilk haritalar, yaklaşık MÖ 2300’lerde Babilliler tarafından kil tabletler üstüne çizilmiştir. Bunlar yeryüzü şekillerinin ilk grafik gösterimleridir. Ama bundan da önce, henüz yazının geliştirilmediği en eski çağlarda insanların aralarında anlaşmak amacıyla saz, taş gibi doğal malzemelerden yararlanarak bazı krokilere ya da planlara başvurdukları sanılmaktadır. Eski çağlardan kalan bazı mağaralarda rastlanan çizimler de bu sanıyı güçlendirmektedir.

Öte yandan 1840’larda güney denizlerini dolaşmakta olan Charles Wilkes’a yol göstermek isteyen adalı bir yerlinin, geminin kaptan köprüsünün güvertesine Tuamotu Adalarının bir haritasını çizdiği bilinmektedir. Eskimolar da yaşadıkları yörenin haritasını, herhangi bir ölçüm aygıtı kullanmamakla birlikte çok doğru bir

biçimde hayvan derileri ya da kemiklerin üstüne çizebilmektedirler. Aynı şeyi İnkalar taş üstüne yapmışlar, üstelik yükseklikleri ya da girinti ve çıkıntıları göstermek üzere taş yontma tekniklerine başvurmuşlardır. Babillilerin yaptığı türden kil tablet haritalarını Mısırlılar da gerçekleştirmiştir. [3]

Haritacılık alanında ilk ciddi adımı Eski Yunanlı coğrafyacı filozoflar atmıştır. İÖ 500'lerde Miletoslu Hekateios bilinen ilk coğrafya kitabını yazmıştır. Ondan bir kuşak sonrada tarihçi Herodotos gezip gördüğü bir çok yeri anlatmış, çizimlerini yapmış ve bu dünyanın o dönemlerde bilinen bölgelerinin haritasını çıkarmıştır.

Önce Pythagoras (İ.Ö. 6.yy) ardından da Parmenides (İ.Ö. 5.yy) Yer'in küre biçiminde olduğu yolunda düşünceler geliştirdiler. İ.Ö. 350'de ise Aristoteles Yer'in küresel olduğu yolunda altı kanıt gösterdi. Onun öğrencilerinden Dikaiarkhos, dünya haritasının üstüne, Cebelitarık ile Rodos'tan geçen doğu-batı doğrultusunda bir yönelim çizgisi çizdi. Kyreneli Erasthenes, Surlu Marinos ve Ptolemaios referans çizgileri sistemini geliştirdiler. Sonunda paraleller ve meridyenler ortaya çıktı, ayrıca bunların izdüşümlerinin elde edilmesi için teknikler bulundu. Eski dönem coğrafyacıları ve haritacıları içinde en önemli etkiyi, İskenderiyeli astronom ve matematikçi Claudius Ptolemaios (İS 90-168) yaptı. Sekiz ciltlik Geographike Hyphegesis adlı yapıtında Ptolemaios, harita izdüşümleriyle küre yapımına ilişkin temel ilkeleri geliştirdi; ayrıca 8000 bölgenin adlarını, enlemlerini, boylamlarını verdi. [3]

Hristiyanlığın başlarında, başka bilimsel alanlarda olduğu gibi haritacılıkta da ciddi bir gerileme oldu. Kilisenin emriyle coğrafya kitaplarının ve haritaların çoğu yakıldı, bu alanlarda çalışanlar baskı altına alındı. Ortaçağ boyunca bütün haritalar kilisenin denetimi altında, dogmalara ve kutsal metin yorumlarına dayanarak yapıldı.

Ortaçağda Hristiyan dünyasında haritacılık çok yavaş ilerlerken, İslam ülkelerinde ve Çin'de önemli gelişmeler gösterdi.

Araplar Ptolemaios'un yapıtlarını kendi dillerine çevirdiler ve çalışmalarını onun temelinde sürdürdüler. Pusulayı Avrupalılardan çok önce kullanmaya başlayan Bağdat'lı astronomlar ise tutulum dairesinin (ekliptik) yatıklığını incelediler ve yer meridyenlerinin belirli bölümlerini ölçtüler. Çinliler ise birçok başka alanda olduğu gibi haritacılıkta da Batı dünyasından bağımsız bir gelişme gösterdiler ve ülkelerinin ayrıntılı haritalarını hazırladılar. [3]

1. ve 2. Dünya Savaşları haritacılık üzerinde büyük etkiye bulundu. 1940'tan önce Yer'in yalnızca yüzde 10'unun ayrıntılı haritası bulunurken, bu tarihten itibaren hava fotoğrafçılığı tekniklerinin de gelişmesiyle bu oran her geçen gün arttı. [3]

2.1.3 Harita Dizaynı

Haritaların dizaynı, harita metodunun seçimi ve grafik değerlerin büyüklük farklılıkları, rengi, yön ve şekillerinin belirlenmesi ile ilgilidir. Bu seçimler harita dizaynını meydana getirir.

Harita baskısının siyah-beyaz mı yapılacağına, ekonomik amaçlar ve çoğaltma olanakları içinde karar verilir. Siyah-beyaz haritalarda çok fazla genelleştirme olacaktır.

Harita yazıları için çeşitli stiller ve büyüklükler uygulanabilir.

Harita tasarımını etkileyecek bir diğer konu da onun kağıt üzerinde mi dijital ortamda mı hazırlanacağıdır. Bilgisayar her konuda olduğu gibi, haritanın hem oluşturulması hem de kullanımı aşamalarında avantajlıdır. Kağıt haritalarda sembollerin darmadağın olmasından çekinilir, ancak elektronik haritalarda, harita belli ölçeklerde iken bazı yazılar gösterilip, bazı tür eleman isimlerinin ise harita yaklaştırıldığında gösterilmesi gibi ayarlamalar yapılabilir.

Özel renk seçimleri, sonucun ifade edilmesini, haritanın anlaşılabilirlik derecesini ve sınırlı bilginin üslubunu etkileyecektir.

Haritaları kimin göreceği ve kullanacağı düşünülerek tasarım yapılması uygun olur, yani kullanıcı kitlesi belirlenir. Harita kullanımının verimli ve sorunsuz olması için kullanıcıların deneyimleri ve onların haritalar hakkındaki ön bilgileri değerlendirilebilir.

Haritanın geçerliliği için, harita ile iletişimde iki değişik yaklaşım yapılabilir ve harita buna göre üretilir. Birincisi, haritanın eğitilmiş kullanıcı tarafından okunacağı göz önüne alınarak dizayn edilmesidir veya ikincisi deneyimsiz kullanıcıya göre harita dizaynı yapılmasıdır. Burada önemli olan iletişimdir, yani kullanıcı kitle ile aynı grafiksel dili konuşmaktır. Örneğin, hedef kitle belli yaşlar arasındaki öğrenciler ise bu yaş grubundakilerin atlaslardaki kaç tematik harita arasında bağlantı

kurabildikleri gibi bir bilgi, bu kitle için hazırlanacak bir harita dizaynında yol gösterecektir.

2.1.4 Harita Çeşitleri

2.1.4.1 Fonksiyonlarına Göre Harita Çeşitleri

İnsanlar yol haritalarını bir yerden diğer bir yere gidecekleri rotayı çizmek için kullanırlar ve yolculukları boyunca rotalarını haritaların yardımıyla kontrol edebilmeyi isterler.

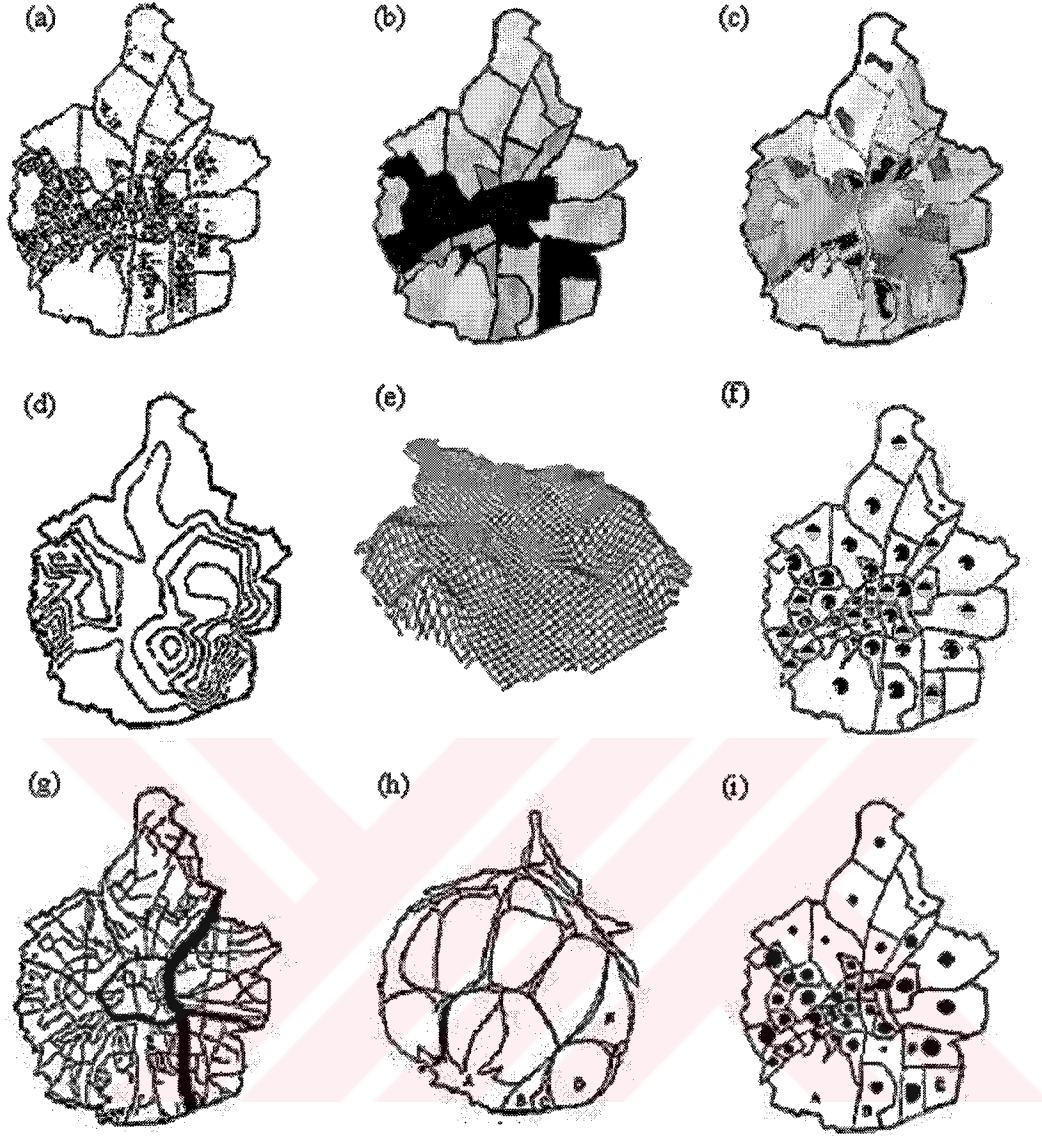
Yol haritalarına göre daha az kullanılanlar fiziksel planlama için kullanılan haritalardır. Fiziksel planlamada kullanılmak üzere, değişik haritalar bulunabilir.

Haritalardan yönetim amaçları için kullanılanlar da vardır, bunlar genellikle büyük ölçekli haritalardır. Bu haritalar yönetimler için yapılır ve ana objeler örneğin yollar, demir yolları, ormanlar, setler, kanallar ve hava alanları vs. gösterilir.

Eğitimsel kullanım için özel amaçlı harita malzemeleri ilk defa 1950'li yıllarda üretilmiştir. Okul atlasları, duvar haritaları gibi.

2.1.4.2 Oluşturulma Metotlarına Göre Harita Çeşitleri

Haritalar, oluşturulma metotlarına göre sınıflandırılabilir. Aşağıdaki şekilde değişik metotlar kullanılarak oluşturulmuş harita örnekleri vardır. Bu şekil 2.1'de (a) nokta gösterimli haritaya bir örnektir, yoğunluğa bağlı olarak nokta sayısı artmaktadır. (b) renk tonlu harita gösterime bir örnektir, bu haritalarda genel olarak renk tonunun koyulaşması yoğunluğun arttığı gösterir. (c)'de keyfi renklendirme ile gösterim yapılmıştır, (d) eşdeğer çizgileri oluşturma metoduna bir örnektir, (e)'de istatistiksel yüzey oluşturulmuştur, (f) harita üzerinde diagram gösterimine (g) ağ ve akış çizgileri ile ifadeye bir örnektir, (h) kartogram metodu, (i) oransal nokta sembolü ile gösterim metodu ile gösterime bir örnektir. [4]



Şekil 2.1 Oluşturulma metotları farklı çeşitli haritalar görülmektedir. [4]

2.1.4.3 Konularına Göre Harita Çeşitleri

Haritalar konularına göre de ana gruplara ayrılabilirler. Jeolojik haritalar, nüfus haritaları, dil haritaları, hava durumu haritaları vb. gibi.

2.1.4.4 Ölçeklerine Göre Harita Çeşitleri

Bilgisayar teknolojisinin sağladıkları düşünüldüğünde akla, 'yeni teknoloji uygun ölçek seçiminde yaşanan sorunları ortadan kaldırdı mı?' sorusu gelmektedir. Bir kere sayısallaştırıldıktan sonra, aynı haritayı bilgisayar teknolojisi ile kolayca büyütebilir ve küçültebiliriz. Ancak, bu konunun üzerinde durulmalıdır. Çünkü, konumsal verinin sayısallaştırıldığı ölçek, gerçekten de önemlidir. Harita verisi belirli bir

ölçekte sayısallaştırıldığında ve çok daha küçük bir ölçekle ekranda görselleştirildiğinde, harita üzerindeki konumsal veri çok kaba olabilir, ayrıca bu pek çok kayba sebep olur. Bazı detaylı karaktere sahip özel nesneler, mesela kıyı çizgileri, küçük ölçeğe indirildiğinde, karmaşık bir çizgi olarak ortaya çıkabilir. Diğer yandan, belli bir ölçekte sayısallaştırılan çizgi elemanları ekranda çok daha büyük gösterildiğinde, gerçeği doğru yansıtmaz. Düzgün çizgiler ile keskin açılar gibi karakteristikler küçük ölçeklerde kabul edilebilirdir ancak, büyük ölçeklerde doğal gözükmecektir. Bundan dolayı, ihtiyaca göre sayısallaştırılan sınır dosyaları kullanılabilir veya mevcut dosyalar genelleştirilir.

Büyük ölçekli harita kullanımı gerektiren çalışma alanlarına kadastro haritaları ve bazı iş alanları örnek verilebilir. Arazi aitliği ile ilgili kayıtların işlendiği kadastro haritaları büyük ölçekli haritalardır. Ağ yapısı olan bazı iş alanlarında da büyük ölçekli haritalar kullanılır. Bu tür iş alanlarında ayrıntılı haritalar bulundurmamak gerekecektir. Örnek olarak gaz, su, kanalizasyon, elektrik, telefon, kablolu televizyon ağları içeren haritalar verilebilir.

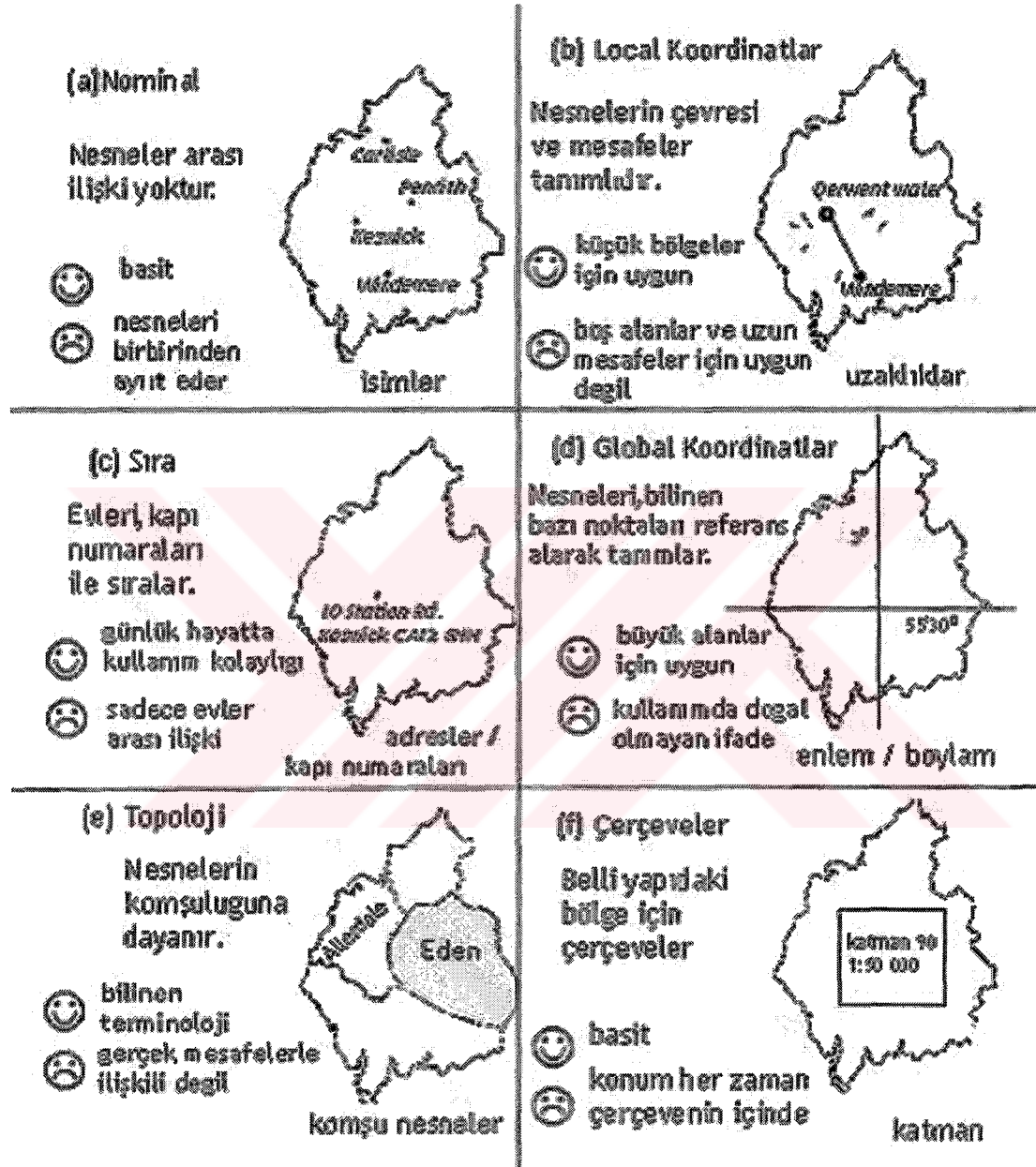
Küçük ölçekli harita kullanımı gerektiren alanlar için sosyo-ekonomik haritalar ve çevresel haritalar örnek verilebilir. Sosyo-ekonomik haritalarda gösterilen konular, çoğunlukla, nüfus sayımı, yerleşim, işçilik, ticaret, tarım gibi konulara bağlı istatistiklerden elde edilir ve konuları itibarıyla küçük ölçekli haritalarda sunulmaları daha uygun olur. Bunun yanı sıra küçük ölçek haritalar çevresel haritalarda kullanılır. Bu haritalar, bitki örtüsü, toprak, hidrolik, jeoloji, jeofizik ve ormancılıkla ilgili kayıtlar sunabilir. Ayrıca doğal kaynakların kullanımındaki getiriler, kötü kullanımı sonucu meydana gelen kayıplar veya kirlilik haritaları örnek verilebilir.

2.1.5 Haritada Konum Bilgisinin Sağlanması

Coğrafi verinin asıl belirleyici niteliği, konumsal bileşenidir yani nesnenin konum bilgisidir. Tüm coğrafi verileri birbirinden ayırt eden en önemli özellik konum bilgisidir. Bugüne kadar, konum belirtmek ve tanıtmak için pek çok metod geliştirilmiştir.

2.1.5.1 Coğrafi Konumlandırma Metotları ve Karşılaştırmaları

Coğrafi konumlandırma için bir çok metod kullanılmaktadır. Her bir metodun farklı avantaj ve dezavantajları vardır. Şekil 2.2’de bu metotlar gösterilmiş ve karşılaştırmak bakımından her birinin avantaj ve dezavantajları belirtilmiştir.



Şekil 2.2 Coğrafi konumlandırma metotları, avantaj ve dezavantajları görülmektedir. [4]

En yaygın konum bilgisi koordinat sistemleri ile verilir. Koordinat sistemlerinde enlem ve boylam kullanılır.

Eğer iki ya da daha fazla veri kümesini birleştirerek çalıştırmak ve bir konumsal analiz yapmak istiyorsak bu kümelerin hepsinin aynı koordinat sisteminde tanımlı olması veya birleştirilmesi gerekir.

Ancak pratikte bu konuda bazı sorunlar yaşanmaktadır. Hangi metot ve tekniklerin, hangi tür problemlerin çözümü için uygun olacağı araştırılmıştır. Ayrıca bir sistemi diğer bir sisteme dönüştürmek için uygulanan bazı metotlar vardır. Bu bilgileri bilmeden sağlıklı ve güvenilir sonuçlar alınamaz, haritaların doğruluğuna güvenilemez.

2.1.5.2 Coğrafi Koordinat Sistemi, Enlemler, Boyamlar

Dünyanın hemen hemen ideal bir küre olduğu temeli üzerine kurulu bir sistemdir. Bu sisteme coğrafi koordinat sistemi denir ve konumlar paraleller (enlemler) ve meridyenler (boylamlar) ile tanımlanır.

Bu küre sisteminin merkezi, ekvator ve Greenwich başlangıç meridyeninin kesişimi kabul edilir. Bir meridyen ile kesişen tüm tam daire düzlemlere paraleller denir ve kuzey kutbundan güney kutbuna uzanan tüm yarım daire düzlemlere de meridyenler denir. Paraleller için Ekvator 0° kabul edilir ve bu durumda kuzey kutbu $+90^\circ$ ve güney kutbu -90° 'dir. Greenwich, 0° meridyen olarak düşünülür ve doğusu $+180^\circ$ 'e ve batısı da -180° 'e kadar derecelendirilir.

Bir konum tanımlamak için kullanılan enlem (ϕ) ve boylam (λ), dünyanın merkezinden yüzeyine ölçülen açılarla tanımlanır. Daha hassas olarak ise dakika cinsinden de ifade edilebilir, $60^\circ 10' N$ ve $24^\circ 58' E$ gibi.

Tüm meridyenler boyunca 1° , 111 km'ye denk gelir.

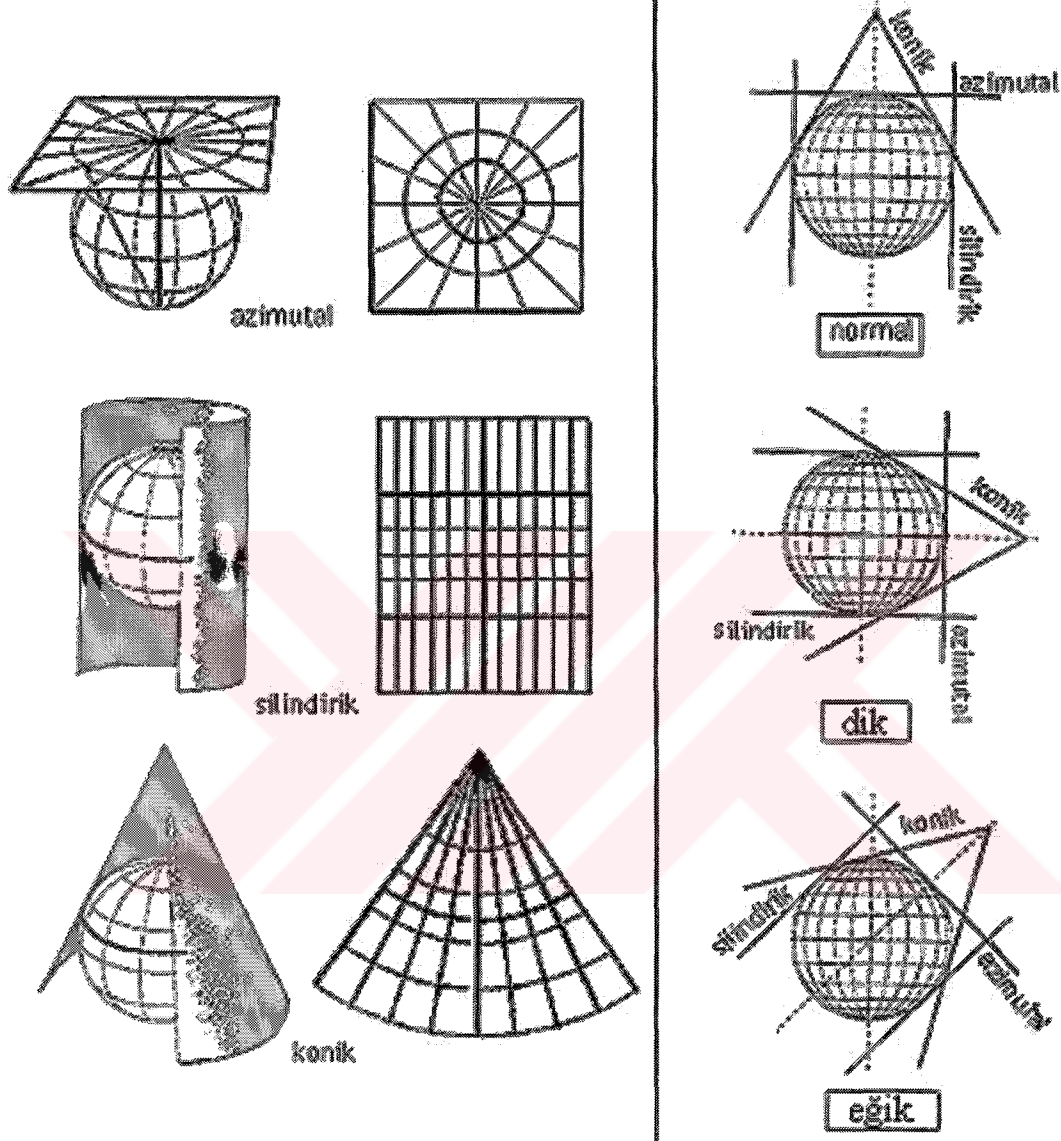
2.1.5.3 Harita Projeksiyonları

Harita projeksiyonları, projeksiyon düzleminin şekline göre sınıflandırılabilirler. Bu durumda konik, silindirik, azimutal projeksiyonlar olarak 3 sınıftan bahsedilecektir.

Elipsoide projeksiyon düzleminin değdiği nokta (ya da çizgi), eğim noktası (ya da çizgisi) olarak adlandırılır. Bu noktada (ya da çizgi boyunca) bozuntu yoktur.

Azimutal projeksiyonların bozuntusuz olan tek bir noktası vardır. Silindirik projeksiyonlarda bir çizgi boyunca bozuntu yoktur. Paralel ve meridyenler birbirlerine diktir ve kutuplara gidildikçe bozuntu artar. Konik projeksiyonlarda da

yine bir çizgide bozuntu yoktur. Silindirik ve konik projeksiyonlarda bir tane eğim çizgisi vardır ve bundan uzaklaştıkça bozulma artar, bu nedenle, bazı projeksiyon düzlemlerinde iki tane bozuntusuz çizgi kullanılarak hazırlanan düzlem kesiştirilir ve böylece bozulma azaltılmaya çalışılır.



Şekil 2.3 Azimutal, silindirik ve konik projeksiyonlar ile bunların normal, dik ve eğik yerleştirilmeleri [4]

Şekil 2.3'te, sol tarafta azimutal, silindirik ve konik olarak düzlemlerin nasıl kullanıldığı, sağ tarafta da bu düzlemlerin 3 türde yerleştirilmesi gösterilmektedir.

Tüm bu projeksiyonların kendilerine özel bozuntu durumları vardır. Mümkün olan en az bozulmaya sahip olmak için, çalışmaya en uygun olan harita projeksiyonu seçilecektir.

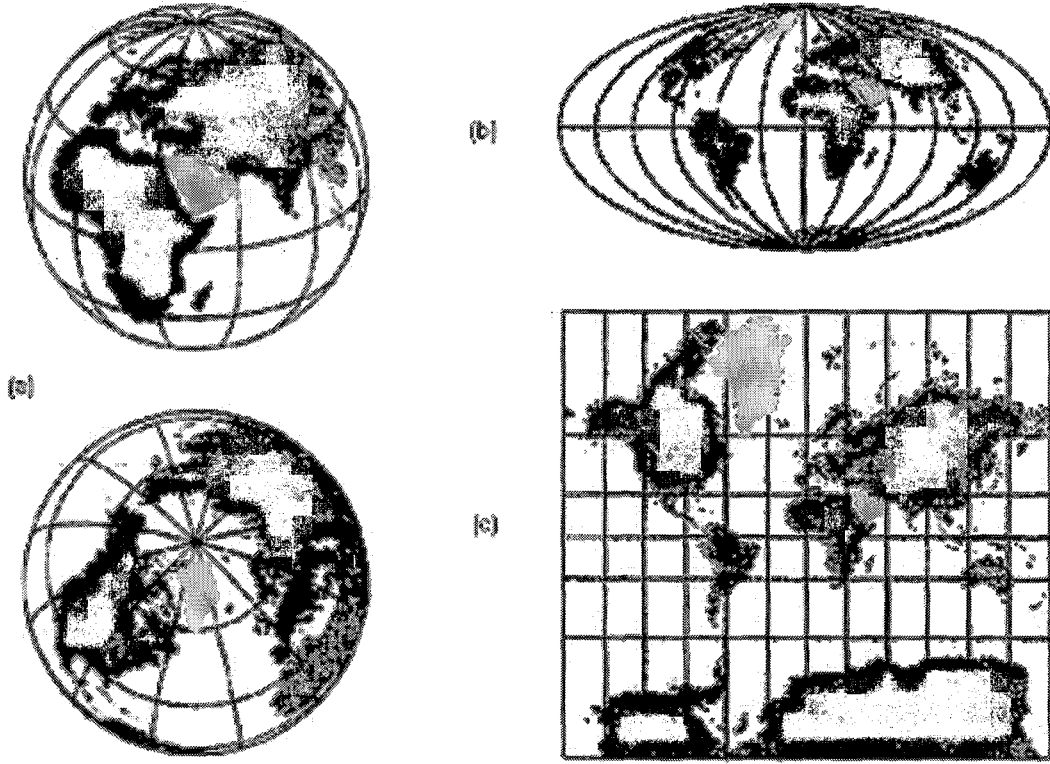
Tüm konumsal işlemler, ancak verilerin aynı projeksiyona sahip olmaları ve birleştirilebilmeleri ile yapılabilir.

Küresel coğrafi koordinatları, 2 boyutlu düzleme çevirmek üzere matematiksel formüller kullanılmaktadır. Bu çevrim sürecinde bir projeksiyon sistemi esas alınır. 3 boyutlu elipsoidi, bazı bozuntular olmadan 2 boyutlu düzleme çevirmek mümkün değildir. Bu bozuntudan şekiller, uzaklıklar ya da doğrultular etkilenebilir. Bozuntulardan hangisinin önemli olmadığı veya daha tercih edilebilir olduğu, ilgilenilen alan ya da kullanım amacına bağlı olarak değişecektir.

Hangi harita projeksiyonunun kullanılacağına, yapılacak çalışmada hangi tür bozulmanın önemszenmeyeceğine göre karar verilir. 3-boyutlu küreden 2-boyutlu düzleme geçilirken, sadece bazı karakteristikler korunabilir, hepsi aynı anda korunamaz.. Korunabilen bu karakteristikler nesne şekilleri, alan boyutları ya da mesafeler olabilir. Buna göre harita projeksiyonları şekil koruyan, alan koruyan, veya mesafe koruyan olarak sınıflandırılır.

Eğer haritalanarak gösterilecek olayın veya oluşumun doğası, nesne şekillerinin korunmasını gerektiriyorsa, şekil koruyan harita kullanılır. Şekil koruyan haritalar sadece küçük alanlar için yapılabilir ve tüm yeryüzü için yapılması mümkün değildir. Bir şekil koruyan projeksiyon, yerel şekillerin aynı kalmasını garanti eder, yani bir küre üzerine çizilen dairesel şekiller, iki boyutlu düzlemde de dairesel olacaktır ancak dairelerin büyüklüğü değişebilir ve küre üzerindeki yerine ve eğim çizgisine veya noktasına göre bozulma olacaktır. Eğer alanlar korunmak isteniyorsa, alan koruyan projeksiyon seçilmelidir. Bir daire düzlemde de, küre üzerinde de aynı alanı kaplayacaktır. Ancak şekli düzlem üzerinde biraz bozularak mesela elipse dönebilecektir.

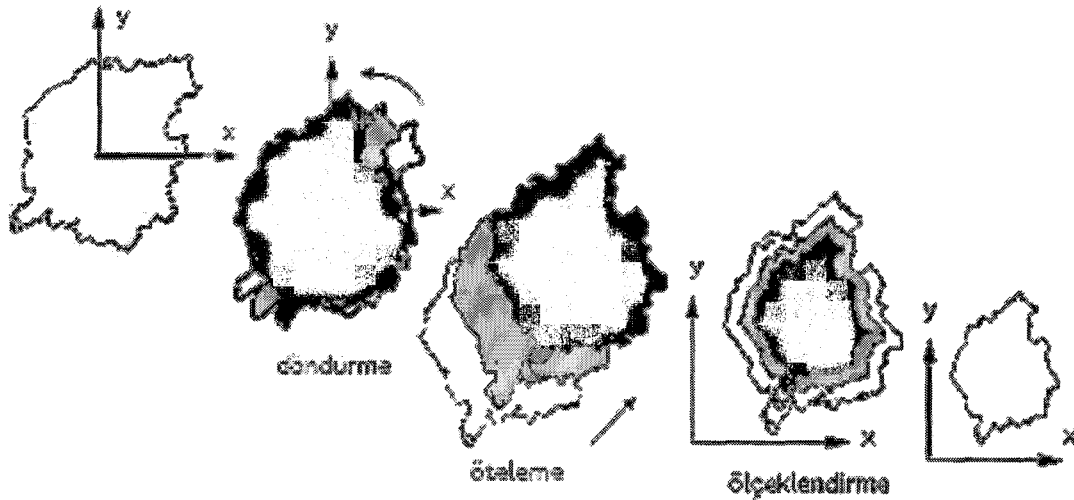
Grönland ve Suudi Arabistan hemen hemen aynı yüzölçümüne sahiptirler. Küre üzerinde görünüşleri, şekil 2.4 (a)'da gösterildiği gibidir. (b)'de alan koruyan harita projeksiyonu kullanılmış ancak şekiller bozulmuş, (c)'de ise şekil koruyan projeksiyon kullanılmıştır, ve alanlar üzerindeki etkisi görülmektedir. [4]



Şekil 2.4 Grönland ve Suudi Arabistan'ın alan koruyan ve şekil koruyan projeksiyonlarla gösterimi. [4]

Geometrik Transformasyon ile bir koordinat sistemi, bir diğer koordinat sistemine dönüştürülür. Bu dönüşümü uygulamak için eski ve yeni koordinat sisteminde bilinen 3 nokta kullanılır.

Bir transformasyon döndürme, öteleme ve ölçeklendirme işlemlerinin birleşimidir. Şekil 2.5'te geometrik transformasyon işlemleri görülmektedir.



Şekil 2.5 Geometrik transformasyon işlemleri görülmektedir.

2.2 COĞRAFI BİLGİ SİSTEMLERİ

2.2.1 Konumsal Verinin Dijitalleşmesi ile Haritanın Yeni Ortamı

Konumsal verinin dijitalleşmesi ile haritaların taşındığı ortam değişmiş oldu. Bilgisayar teknolojisi, haritaları ekrana taşıdı. Bu haritalar üzerinde, veritabanı teknolojisi sayesinde sorgulamalar ve bazı fonksiyonlar aracılığıyla değişik işlemler yapılabilmektedir. Seksenli yıllarda bu tür sorgulama ve analizlere imkan veren yazılım paketleri, Coğrafi Bilgi Sistemleri (CBS) adını aldı.

“Coğrafi bilgi sistemleri, coğrafi nesnelere ait coğrafi verilerin toplanması, doğrulanması, depolanması, bu verilerin veri tabanı işlemleri, sorgulamalar, dönüşümler, coğrafi analizler ile bilgiye dönüştürülmesi ve coğrafi veri ve bilgilerin gösterimi için kullanılan gelişmiş bir bilgi sistemidir.” [5]

Kağıt haritalar, konumsal verinin depolanması ve sunumu işlevini üstlenmekteydiler ve hala aynı görev için kullanılırlar. Ancak, dijital ortamdaki haritaların görevleri oldukça genişlemiştir ve sunum ortamları da çoğalmıştır. Artık haritalar, önceden olduğu gibi, sadece sonuçların sunulduğu ortamlar olmaktan çıkmış, analiz aşamaları boyunca da veri ve araç olarak kullanılmaya başlanmıştır. Ayrıca, veri tabanı teknolojisi ve bilgisayar grafiği tekniklerinin kullanılması, üç boyutlu ve canlandırılmalı haritalar gibi yeni ve alternatif sunum olanakları ortaya koymuştur.

2.2.2 Coğrafi Bilgi Sistemleri Bileşenleri

Coğrafi bilgi sistemleri, 4 bileşenden oluşan bir organizasyon olarak tanımlanabilir:

1. Coğrafi Veriler
2. CBS Donanım Bileşenleri
3. CBS Yazılım Bileşeni
4. CBS Personeli

2.2.2.1 Coğrafi Veriler

Coğrafi verilerin toplandığı kaynaklar şunlardır: [6]

- Çizgisel haritalar
- Ortofoto haritalar ve foto haritalar
- Sayısal haritalar
- Hava fotoğrafları
- Uydu görüntüleri
- Arazi ölçmeleri
- Tapu ve kadastro kayıtları
- Bilgisayar destekli tasarım (CAD) çizimleri
- Veri tabanı kütükleri

2.2.2.2 CBS Donanım Bileşenleri

Coğrafi bilgi sistemleri donanım bileşenleri üç grupta toplanır:

1. Veri giriş elemanları
2. Veri depolama ve işleme elemanları
3. Veri sunuş elemanları

Veri giriş elemanları:

- El ile sayısallaştırıcılar
- Otomatik Sayısallaştırıcılar

Veri depolama ve işleme elemanları:

- Bilgisayar belleği (Veri depolama)
- Merkezi işlem birimi (Veri işleme)

Veri sunuş elemanları:

- Ekran
- Yazıcılar

- Çiziciler [6]

2.2.2.3 CBS Yazılım Bileşenleri

CBS yazılım bileşenleri, coğrafi bilgi sisteminden beklenen fonksiyonları yerine getirebilmek için kullanılan bileşenlerdir. Bir coğrafi bilgi sisteminden beklenen fonksiyonlar şunlardır: [6]

- Sayısallaştırma ve editleme
- Yapılandırma
- Genelleştirme
- Raster-vektör ve vektör-raster dönüşümü
- İlişkilendirme
- Pafta işlemleri
- Özet bilgi çıkarma
- Birleştirme, ayırma
- Analiz ve sorgulama

Coğrafi bilgi sistemi yazılımları 6 grupta toplanır:

1. Coğrafi bilgi sistemi veri giriş yazılımları
2. Veri tabanı depolama ve yönetim yazılımları
3. Veri dönüşüm yazılımları
4. Sorgulama yazılımları
5. Coğrafi bilgi sunuş yazılımları
6. Kullanıcı ile iletişim yazılımları

Bu yazılım çeşitleri ayrıntılı olarak incelendiğinde aşağıdaki özelliklere sahip oldukları görülür:

1. CBS Veri Giriş Yazılımları

Coğrafi verileri uygun coğrafi bilgi sistemi veri giriş elemanları kullanarak bilgisayar tarafından anlaşılır formlara dönüştürme işlemlerini yapan ve daha sonra bilgisayar

ortamına aktarılmış verilerin doğruluğunu kontrol etmek ve düzeltmek için kullanılan yazılımlardır.

Gerek grafik gerekse grafik olmayan veri girişleri katman mantığına uyarak, katmanlar şeklinde yapılır.

2. Veri Tabanı Depolama ve Yönetim Yazılımları

Veri tabanı depolama ve yönetimi yazılımları, coğrafi bilgi sistemine aktarılan hatalardan arındırılmış verileri, bir veri tabanı modeline uygun şekilde depolayan ve onların kolayca yönetimini sağlayan yazılımlardır.

Coğrafi veri depolama teknikleri, coğrafi verilerin grafik veri yapılarına bağlı olarak, raster yapıdaki coğrafi veri depolama teknikleri ve vektör yapıda coğrafi veri depolama teknikleri olarak iki kısma ayrılır.

3. Veri Dönüşüm Yazılımları

Coğrafi veri dönüşümü, sadece farklı veri yapıları, farklı veri formatları veya farklı veri türleri arasındaki dönüşüm işlemini gerçekleştirmez. Aynı zamanda veriler üzerinde yapılan değişiklikler ve böylece verilerden yeni veriler elde etmek amacıyla gerçekleştirilen işlemleri gerçekleştirmektedir.

Veri dönüşümü fonksiyonları iki ana gruba ayrılır:

a. Veri bakımı ve idaresi fonksiyonları

b. Veri kullanımı ve analizi fonksiyonları

a. Veri bakımı ve idaresi fonksiyonları aşağıdaki işlemleri kapsar.

Grafik detayların ekleme, silme, değiştirme ile güncellenmesi,

Haritanın ölçeğinin değiştirilmesi, ötelenmesi, döndürülmesi,

Haritanın bir projeksiyon sisteminden başka bir projeksiyon sistemine dönüştürülmesi,

Komşu kenarların kenarlaştırılması,

Görüntüyü uzaklaştırma, yaklaştırma ve kaydırma işlemleri,

Vektör veri yapısından raster veri yapısına dönüşüm ve ters işlemi,

Kartografik genelleştirme, çizgi yumuşatma, yeniden sınıflandırma işlemleri,

Bozuntu giderme işlemi [6]

b. Veri kullanımı ve analizi fonksiyonları:

Harita cebri olarak da adlandırılan ve coğrafi veriyi bilgiye dönüştüren ve üretilen bu bilginin yeni bilgilerin üretilmesinde kullanılmasını sağlayan fonksiyonlardır. CBS'lere kendi içinde doğurganlık kazandıran bir bileşendir. [5]

Çizgi ve yüzey enterpolasyonları, yüzey modellendirme,

İstatistiksel analiz,

Tampon yaratma,

Üst üste çakıştırma,

Yeniden sınıflandırma,

Alan birleştirme,

İkili cebir,

Ağ analizi gibi işlemlerden oluşan coğrafi analiz pek çok CBS projesinin temel hedefini oluşturmaktadır. [5]

“4. Sorgulama Yazılımları

Mekansal sorgulama: Konumu belli olan grafik verinin nitelik bilgilerinin sunulması veya öznitelik bilgileri belli olan coğrafi verinin konumsal bilgilerinin sorulması,

Mekansal analiz

Yakınlık analizi

Mantıksal işlemler

Sınır işlemleri

Mekansal birleştirme

Ağ analizi

Ölçme işlemleri

Sayısal arazi analizi

Mekansal enterpolasyon

Eş yükselti eğrilerinin otomatik çizimi

Eğim belirleme

Gölgelendirme

Görünürlük analizi

Hacim hesabı

Kesit alma

Üç boyutlu görüş [6]

5. Coğrafi Bilgi Sunuş Yazılımları

Çeşitli nitelikteki terminallerden yazıcılar, çiziciler ve manyetik ortama kadar uzanan birimler aracılığı ile değişik ürün haritalar, tablolar ve şekiller üretebilen yazılımlardan oluşur. CBS grafik verisi olmayan klasik bilgi sistemlerinden konuma bağlı ve topolojik ilişkileri içermesi bakımından farklılaşmaktadır. Buna bağlı olarak da analiz ve sorgulama olanakları oldukça zengindir.

Grafik ve grafik olmayan veri girişleri, aynı geometrik özelliğe ve ortak tanımsal özelliklere sahip detaylar için farklı katmanlara yapılır. Belli grafik ve grafik olmayan bilgiler, katmanlar halinde bilgisayar ortamına girilir. [6]

6. Kullanıcı ile İletişim Yazılımları

Coğrafi bilgi sistemi ile kullanıcının etkileşimi için gerekli menüler oluşturan, verilere erişimi sağlayan programlar, programlama dili ve sorgulama dili yazılımlarıdır.

2.2.2.4 CBS Personeli

Bir coğrafi bilgi sistemi personel bileşeni, dokuz grupta incelenebilir:

- Sistem yöneticisi
- Sistem analisti
- Veritabanı yöneticisi
- Veri işleme uzmanı
- Harita mühendisi
- Veri giriş operatörü

- Veri güncelleme elemanları
- Bilgisayar mühendisi
- Uygulama programcısı
- Son kullanıcılar

2.2.3 CBS Yapısı

2.2.3.1 CBS Projesi Aşamaları

CBS proje aşamaları aşağıdaki şekilde maddelenebilir:

- Gerçek dünya asıl veri kaynağıdır. Bu ortamdan modele aktarılabilecek varlıklar nelerdir, hangi coğrafi nesneler ile çalışılacaktır belirlenir. Yaşadığımız çevrede ormanlar, yollar, akarsular, binalar gibi pek çok coğrafi nesne bulunmaktadır. Bu tip varlıklar, sistemde ifade edecekleri anlam bakımından gözden geçirilir. Temel varlıklar sisteme aktarılacaktır.

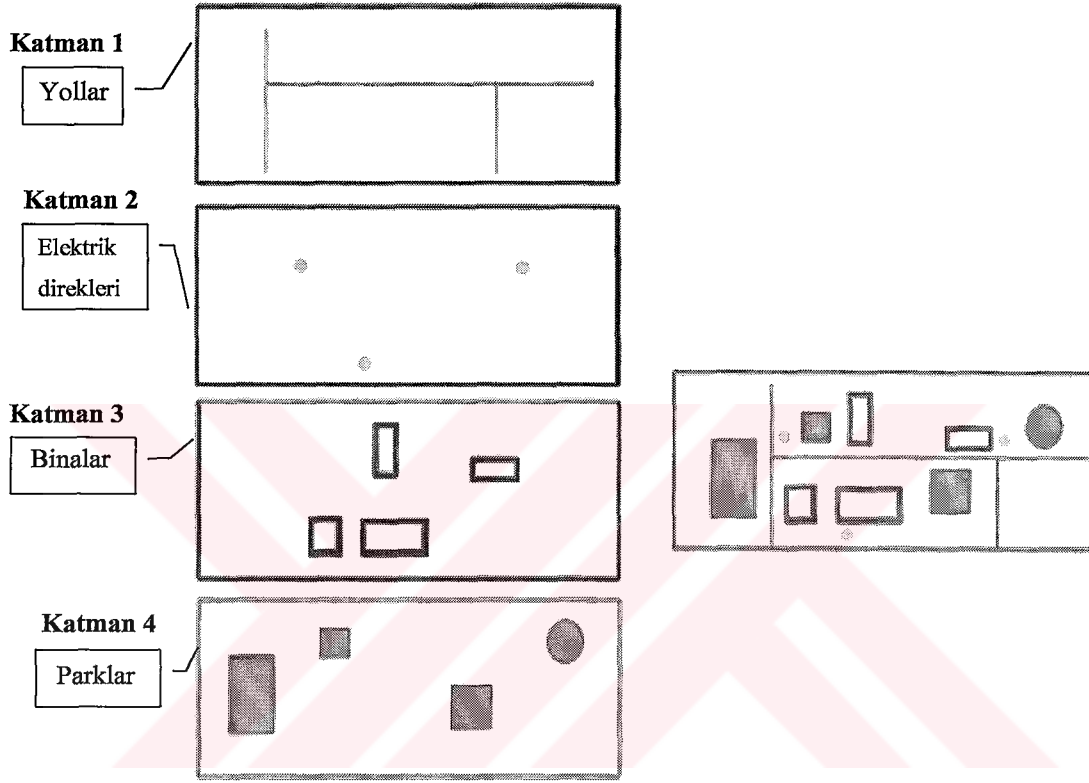
- Sistemde modellenen çevre ve kullanılmak üzere belirlenen coğrafi nesneler hakkında genel bilgi edinilir, veri toplanır.

- Sisteme veri girişi sağlanır.
- Veri yönetimi ile veriler düzenlenir, çalıştırılır.
- Sorgulama ve analiz programları ile veri üzerinde çalışılır.
- Karar aşamasında kullanmak üzere gereken bilgiye ulaşılır.
- Kullanıcılara sunulur.
- Karar sonucu gerçekleştirilir.

2.2.3.2 Katman Mantığı

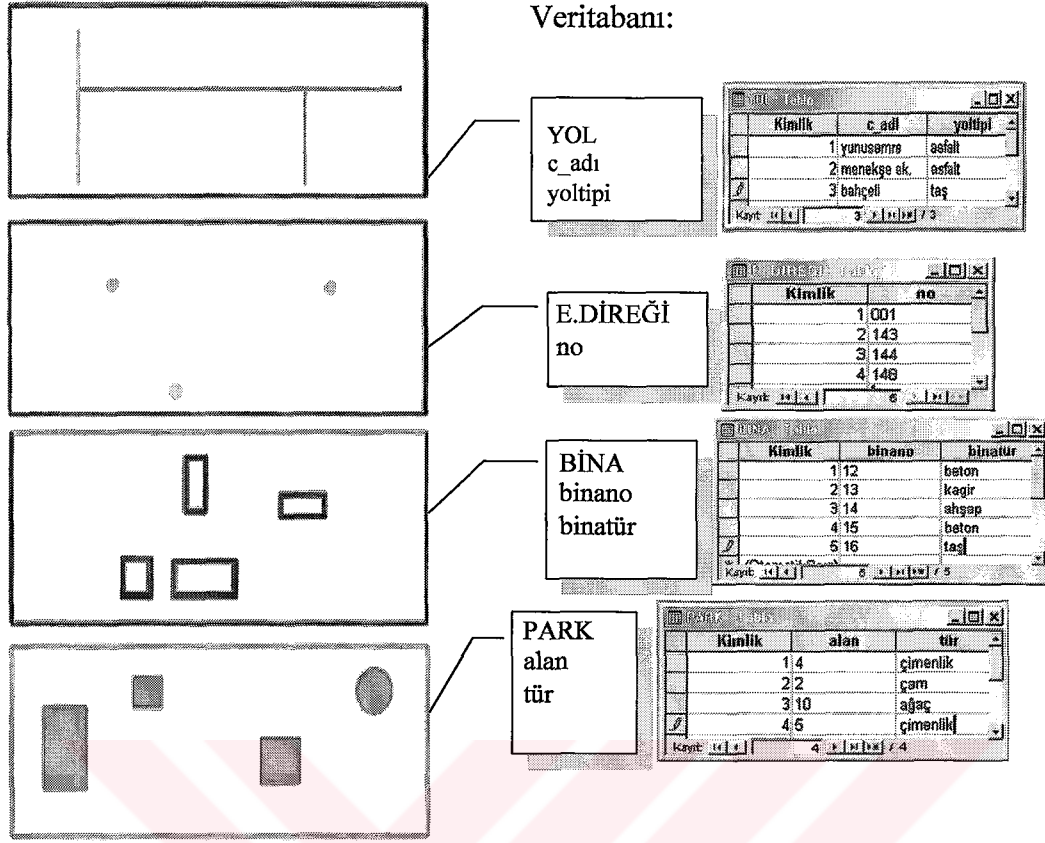
CBS'nin en belirgin ve etkili özelliği katman mantığına dayalı bir yapıda olmasıdır. Veri girişi katmanlara yapılır ve sistem verileri kayıtlı oldukları katmanlar ile tanımlar. CBS'de yapılacak analize göre, gerekli katmanlar gösterime açılarak, kullanılmayacak olanlar kapatılabilir. Şekil 2.6'da, bir CBS projesinde katman mantığının ne olduğu anlatılmaktadır. Basit olarak bir CBS projesi için yollar, elektrik direkleri, binalar ve parklar şeklinde dört varlık tipi belirleyelim. Bunlar o bölge için içerdikleri tüm örneklerle birlikte bir veri kümesi oluştururlar. Bu veri

kümelerinin hepsi için ayrı bir katman belirlenir. Örneğin yollar birinci katmana, elektrik direkleri ikinci katmana, binalar üçüncü ve parklar da dördüncü katmana çizilmiştir. Bu katmanların hepsinin aynı anda gösterimi sağ tarafta görüldüğü gibi olacaktır. Her bir katman bir veri kümesinin tüm örneklerini aynı tarzda gösterir. Bu gösterimi renkler, kalınlıklar, dolgu ve desenler ve kullanılıyor ise semboller belirleyecektir.



Şekil 2.6 Coğrafi Bilgi Sistemlerinde katman mantığı.

Bu katmanların her biri, coğrafi nesnelerin geometrik bileşenini gösteren ortamlar olarak düşünülebilir. Veritabanı kısmında ise, bu katmanların hepsi bir tabloya karşılık gelmektedir. Şekil 2.7’de katman-tablo ilişkisi görülmektedir.



Şekil 2.7 CBS sistemlerinde katman-tablo ilişkisi

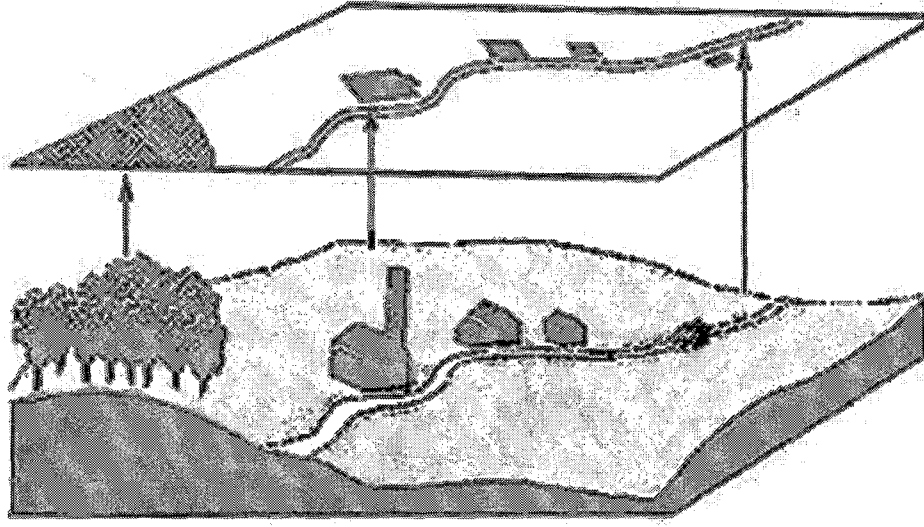
Örneğin binalar katmanı veritabanında binalar tablosuna denk gelmektedir ve o katmandaki her örnek (yani her bina) binalar tablosunda bir kayıta aittir. Bu kayıtlar tablodaki satırlardır ve bir CBS elemanının nitelik bilgilerini ayrı sütunlarda tutmaktadırlar.

2.2.4 CBS Verisi Bileşenleri

CBS ile proje yapmak için ilk önce, konumsal veriyi iyi kavramak gerekmektedir. CBS sistemlerindeki verinin, diğer veri sistemlerinden farklı bir boyutu vardır. CBS projesinde kullanılan veriler, uzayda belirli bir konuma sahip olan verilerdir. Bu nedenle, konumsal adresleri ile görselleştirilirler ve bu görselleştirme bir harita olarak ortaya çıkar.

Gerçek hayattan CBS verisi oluşturmak üzere izlenen yol önemli bir aşamadır. Gerçek dünyaya ait nesneler, 2 boyutlu sayısal ortama belli ölçeklerde küçültülerek aktarılırlar. Bu işlemler gerçekleştirilirken, bazı nesnelerin noktasal, bazılarının çizgisel ve bazılarının da alansal olarak gösterilmeleri uygun olur. Şekil 2.8'de,

gerçek dünyadan bilgisayar ortamına geçilirken nesnelerin nasıl modellendiği gösterilmiştir.



Şekil 2.8 Gerçek dünyadaki coğrafi nesnelerin CBS ortamına aktarımı [4]

Yaşadığımız dünyanın nesneleri, bilgisayar ortamında sayısal bir model olarak gösterilirler. CBS ortamında nesneler türlerine göre depolanır, kartografik modele dönüşür, ve insan zihninde bütünlük kazanırlar.

2.2.4.1 CBS Verisi Geometrik Bileşeni

CBS ortamında, verinin iki temel bileşeninden biri geometrik bileşenidir. Geometrik bileşen, verinin konumsal ve boyutsal bileşenidir. CBS verileri noktasal, çizgisel ve alansal elemanlar ile sembolize edilirler, bu CBS verisinin geometrik bileşenidir, ayrıca geometrik bileşen elemanlarla birlikte koordinat sistemi bilgilerini de saklamaktadır, yani konum bilgisini de içerir.

2.2.4.2 CBS Verisi Niteliksel Bileşeni

Verinin temel bileşenlerinden diğeri niteliksel bileşenidir ve verinin öznitelik bilgilerinden oluşur. CBS ortamında bir coğrafi nesne, ekranda algılanan geometrik eleman ile temsil edilmenin yanında, bir de veritabanında kayıt olarak temsil edilir. Bu kayıtlarda tutulan varlıklara ait nitelik bilgisi, CBS'nin niteliksel bileşenidir.

2.2.4.3 CBS Verisinin 3. Bileşeni Olarak Zaman

Veriyi oluşturan geometrik ve nitelik değerlerinden sonra, zaman değerinin de üçüncü ana bileşen olduğu söylenebilir. Özellikle CBS veritabanlarında, zaman serilerinin gelişmesi verinin zaman bileşeninin önemini daha da arttırmaktadır. [4]

2.2.5 CBS Kullanımı

2.2.5.1 CBS Kullanımı Gerektiren 3 Durum

CBS ortamında görselleştirme üç ihtiyaçtan dolayı yapılmaktadır. [4]

1. Görselleştirme, bilinmeyen ve genellikle işlenmemiş verinin ortaya çıkarılması amaçlı kullanılmaktadır. Birçok uygulamada, örneğin uzaktan algılama verileriyle çalışırken çok fazla veriye ulaşmak mümkündür. Verilerin konumsal analiz operasyonlarında kullanılmasından önce “Veri setinin doğası nedir?” veya “Şu veri setlerinden hangisi, üzerinde çalışılan problemle ilgili örneklerdir?” gibi sorular cevaplanmalıdır. Burada CBS, veri görüntüleyici olarak önemli bir rol oynar.

2. Görselleştirme, bilinen verilerin kontrol edilmesi amacıyla uygulanabilir, kısaca analizlerde kullanılır. Bir planlama ortamında iki farklı veri setinin doğası anlaşılabilir fakat birbirleriyle olan ilişkileri anlaşılabilir (örneğin zemin suyu düzeyi ve yeni bir yolun muhtemel konumu). Bir konumsal analiz operasyonu ile, iki veri seti kombine edilerek, muhtemel konumsal ilişkileri çözülebilir. Bu tür bir operasyonun sonucu gerekirse planlara adapte edilebilir.

3. Görselleştirme sunum amaçlı uygulanır. Konumsal analiz operasyonlarının sonuçları, iyi tasarlanmış haritalarla sunularak geniş bir kitleye anlaşılabilir.

CBS kapsamındaki görselleştirmelerde, ortaya çıkarma, analiz etme ve sunma olarak ele alınan bu üç farklı kullanım alanı düşünüldüğünde, sunum araçlarının gösterdiği gelişimi fark etmek mümkün olmaktadır.

2.2.5.2 CBS Kullanımı ile Cevaplanabilen Sorular

CBS'nin çözdüğü sorunların, cevaplayabildiği soruların kapsamı çok geniştir. Bunların yanı sıra, kazandırdığı hız başta olmak üzere sağladığı pek çok yarar vardır.

CBS ile cevap bulunabilecek soru ve sorgulamalar aşağıdaki şekilde özetlenebilir [4]:

- Burada ne var . . . ?

Kimliklendirme: Harita üzerinde bir noktaya işaret edildiğinde, orası ile ilgili isim veya depolanmış herhangi başka bir bilgi ekrana gelir.

- . . . nerededir?

Konum: Bu sorunun cevabını veren bir veya birden çok konum olabileceği hesaba katılmalıdır. Burada ayırım, ya bir koordinat setiyle ya da spesifik bir nesnenin konumunu gösteren bir haritayla yapılabilir.

- . . . -den beri ne değişti?

Eğilimler: Bu soru, konumsal verinin zaman bileşenini içermektedir. 1950 ile 1990 yılları arasında aynı şehrin ne şekilde büyüdüğünün CBS ortamında gözlenebilmesi örnek verilebilir.

- . . . arasında en iyi rota nedir?

Optimal yol: Karayolları veya kanalizasyon sistemleri ağlar üzerine kurulmuştur ve CBS ortamında, en kısa veya en ucuz rota nedir şeklinde sorgulamalara cevap aranabilir.

- . . . arasında ne gibi bir ilişki vardır?

İlişkiler: Bu tür sorular karmaşıktır ve birçok veri seti içerirler. Cevaplar, bazı ilişkilendirmeleri açıklayacaktır.

- . . . -sa ne olur?

Modeller: Bu sorular planlama ve tahmin aktivitelerine yöneliktir. Bir örnek olarak şehrin kuzeye doğru büyümesi halinde, yerel toplu taşıma ağı ve kapasitesinin nasıl etkileneceği modellenebilir.

Bir CBS yazılımı ile cevaplanabilen sorular, sorgulama tiplerine göre de sınıflandırılabilir. Bir CBS'nin uzun hesaplamalara gerek kalmaksızın cevaplayabileceği soruların kapsamı o CBS'nin sorgulama uzayıdır ve bu uzayın gerek genişliği gerekse cevap-süre performansı büyük ölçüde verinin depolandığı mantıksal ve fiziksel coğrafi veri yapısının bir fonksiyonudur. Sorgulama uzayı sonuçta bir sorgu listesidir. [5]

Bu sorgulama tipleri şu gruplar altında toplanabilir:

- Grafikten veri tabanına sorgulamalar,
- Veri tabanından grafiğe sorgulamalar,
- Metrik sorgulamalar,
- Topolojik sorgulamalar,
- Düzen sorgulamaları,
- Veritabanı sorgulamaları. [5]

2.2.5.3 CBS Kullanım Alanları

Yönetimlerde karar destek sistemi olarak CBS kullanımına geçilmesi pek çok fayda sağlamıştır. “Kullanılan sistem içinde bilgisayar teknolojisinin (donanım ve yazılımlar) yeri yoksa ve insanlar sadece analog verilerle iş üretiyorlarsa, burada fazla insan gücü istihdamından, gereksiz bürokratik işlemlerden, fazla miktarda arşiv evraklarından, gereksiz veri tekrarından v.b. kaynaklanan zaman ve parasal kayıptan söz etmek yanlış olmaz.” [7]

Coğrafi verilerle uğraşan kurum ve kuruluşlar, uğraş alanları içindeki verileri toplamak, bir araya getirmek, bazı dönüşümlerle birleştirmek ve güncelleştirilmek istediklerinde, güncelleştirilen verileri mevcut haritalara işlemek, yeniden arşivlemek, ya da arşivlerde saklanan bu verilerle sözel verileri birleştirerek bir coğrafi analiz yapmak istediklerinde pek çok kişinin, uzun zaman harcayarak bu konuda çalışmalar yapmaları, istenen bilgileri üretmeleri ve buna göre karar vermeleri gerekir ve hatta bu konuda birkaç değişken üzerinde oynanarak değerlendirme yapılacak ve karara varılacak ise, aynı işlemi birkaç kez de tekrarlamak gerekecektir. Bu çok güç ve çok da kayba sebep olacak bir sistemdir.

1970’li yıllara kadar coğrafi varlıklar klasik aletlerle arazide ölçülür, defterlere yazılır, ölçümlere göre günler süren hesaplamalar yapılır ve bunların sonucuna göre detaylar haritalara manuel olarak çizilirdi. Ya da çok büyük alanlarda hava fotoğrafları analog fotogrametrik aletlerle değerlendirilir ve çizgisel haritalar üretilirdi. 70’li yıllardan itibaren manuel olarak üretilen bu haritalar sayısal ortama aktarılmaya başlandı. 80’li yıllardan itibaren ise coğrafi alanı ölçmeye yönelik sayısal jeodezik aletler, sayısal fotogrametrik sistemler, sayısal uydu görüntüleri ve görüntü işleme yöntemlerinin geliştirilmesi ile vektör ve raster verilerin bütünleşik olarak kullanılması olanağı, bu karmaşık sorunların çözümünde CBS’lerin önemini

arttırmıştır. CBS'lerin en yararlı tarafı coğrafi analiz sonuçlarının anında ekranda görüntülenmesidir. Bir coğrafi alanda eski ve mevcut verilerle mevcut koşullarda tasarlanan modelin, ölçüler değiştiğinde ya da koşulların değişeceği varsayıldığında, model tasarımının da etkileşimli olarak değiştiği ekranda gözlenir. [7]

Pek çok konuda bir araç olarak kullanılacak olan CBS, hem hız, hem öngörü sağlamak için mükemmel bir araçtır. Coğrafi bilgi sistemleri, coğrafi verilerin söz konusu olduğu her alanda uygulanabilir bir yapı sunmaktadır. Coğrafi verinin tanımının ne kadar geniş olabileceği hatırlanırsa, CBS uygulama alanlarının da o denli çok olacağı sonucuna varılır. Bir inceleme CBS teknolojisinin kullanılmakta olduğu 9 temel uygulamanın varlığını ortaya koymaktadır [5]:

1. Tesis ve demirbaş envanteri : Kaynakları en uygun kullanmak amacı ile yer yüzeyinin üzerine, üstüne ve altına dağılmış olan nesnelerin konumlanması, sayımı ve dağılım analizi. Örneğin orman amenajmanı, kadastral parseller, altyapı ağı yönetimi gibi uygulamalar.

2. Coğrafi veri toplama ve üretimi : Uzaysal veri tabanları kurmak ve yaşatmak üzere coğrafi verilerin toplanması. Örneğin elektronik kontrol, mühendislik ve arazi ölçmeleri, sayısal harita üretimi, fiziksel ve kültürel olguların uzaktan algılanması gibi uygulamalar.

3. Harita ve plan basımı : Baskı kalitesinde harita ve planların üretimi. Örneğin planimetrik, topografik, deniz, hava ve tematik haritaların ve diğer benzeri kartografik ürünlerin tek başlarına dağıtım için ya da diğer basılı veya elektronik dokümanlar içerisinde yer almak üzere üretimi.

4. Kaynak Tahsisi : Doğal ve insan yapısı kaynakların politik, ekonomik veya sosyal kriterlere göre tahsisi için konum, kalite, sayı ve hareketlerin analizi. Hedef pazarlama, satış bölge planlaması, hizmet ağı dağıtımı, öğrenci yerleştirme gibi uygulamalar.

5. Rota ve akış optimizasyonu : İnsanların, malların ve hizmetlerin akışının optimizasyonu. Hizmet ağları kapasite yönetimi, ulaşım ağı analizi, okul servis güzergahlarının yönetimi, dağıtım ve toplama araçlarının güzergah ve zamanlama yönetimi gibi uygulamalar.

6. Rota Seçimi ve Navigasyon : Saptanmış kriterlere göre bir ağ içinde en uygun güzergahın seçimi. Acil hizmet araçlarının göreve gönderilmesi, tehlikeli madde taşıyan araçların, taksilerin güzergahlarının belirlenmesi gibi uygulamalar.

7. Tesis konum planlaması : Tesisler için en uygun yerlerin saptanması. İtfaiye ve polis karakollarının, fabrikaların, alışveriş merkezlerinin ve tehlikeli atık yerlerinin seçimi gibi uygulamalar.

8. Yer altı ve yerüstü değerlendirmeler : Doğal kaynakların tespiti, korunması ve en avantajlı kullanımı için yer altı ve üstündeki fiziksel olguların analizi. Topografik, hidrolojik, jeolojik, meteorolojik, jeofizik ve manyetik anomali modellendirmeleri gibi uygulamalar.

9. İzleme ve gözleme : Tamamlayıcı veya düzeltici tedbirler geliştirmek üzere üzerinde çalışılan süreci anlamak için tekrarlı olayları kaydetmek ve analiz etmek. Reklam kampanyası sonuçlarının izlenmesi, seçim, suç, trafik kazaları ve çevre analizi.

2.2.5.4 CBS Kullanımında Mevcut Sorunlar

CBS kullanımının yaygınlaşması ile, kartografik kurallardan bihaber olan birçok kullanıcı da kendi haritalarını üretme imkanı bulmuştur. Bu noktada bir sorun doğmaktadır: Bu haritaların etkin olacağının bir garantisi verilemez.

CBS ile harita üreten kişinin, bir haritacı olması gerekmez; ancak harita üretiminde kartografik gösterim kurallarının göz önüne alınması gerekir. Kartografik kurallar üretilen haritaları daha etkin hale getirir. Ancak, bu kurallar CBS yazılımlarının bir parçası değildirler.

Analistler, haritaları kendimiz ve ilgili kitle anladığı sürece bu kuralları dikkate almanın hayati bir anlamı yoktur demektedirler. Analistler, kendi verilerini ürettikleri haritalardan anlayabilirler, fakat bu haritaları başkalarına gösterdikleri anda problemler ortaya çıkmaya başlar. Bu, bir veri ortaya çıkarma ortamında kullanıcının verinin tam doğasını bilememesinden kaynaklanmaktadır. Bu durum, gösterimin yanlış yönlendirebilmesine müsait bir ortam hazırlar.

CBS ile ilgili ortaya çıkan diğer sorunlardan biri de, tam anlamıyla 3 boyutlu analiz aşamasına geçilememiş olmasıdır. “Tüm mekansal bilginin üç boyutlu olmasına karşın bugün bilgi sistemlerinde ele alınan cisimlerin çoğu iki boyutu ile yer

almaktadır. Bugün 3. boyuttaki bilgi sistemlerine yönelik çok sayıda çalışma yapılmasına ve uygulamadan talep olmasına karşın iki boyutlu duruma göre karmaşık ve güç algoritmaların geliştirilmesi gerekmektedir. Bugün bilgisayar destekli animasyon ve çizim konularında çok sayıda üç boyutlu topolojiye dayanan uygulamalar olmasına karşın bir bilgi sisteminin sorgulama yapısına dayanan katmanlara ayrılmış üç boyutlu bilginin biriktirilmesi ve sunuşa hazır hale getirilmesi hala üzerinde çalışılan konudur.” [8]

2.2.5.5 Harita ve CBS’nin Geleceği

Haritanın geçmişten günümüze uzanan gelişimine bakıldığında, geline nokta çok daha iyi anlaşılmaktadır. İlk elektronik atlaslar sıralanmış slayt gösterileri şeklindeyken, gelişmiş elektronik atlaslar yüksek interaktif haritacılık kapasitesine sahipler ve kullanıcıya kendi verileriyle atlas verilerini beraber kullanma şansını vermektedirler. Fazla içerikli konumsal verinin gösterimi ihtiyacı bilimsel görselleştirme, multi-medya, sanal gerçeklik ve keşfe yönelik veri analizleri aracılığı ile karşılanmaktadır. Kullanıcı metin, haritalar ve hatta sesli video görüntüler gösteren çok pencere ekranlarla yüz yüze gelmektedir.

CBS’nin gelişimine bakıldığında ise kullanımının gün geçtikçe daha geniş bir yelpazeye dağıldığı görülmektedir. Gelecekte de bu gelişimi sürdüreceği düşünülmektedir.

“Şimdiye kadar kartografya, fotogrametri, jeodezi ve uzaktan algılama gibi ilgili disiplinler ile bilgisayar ve bilgi sistemleri teknolojisi araçlarının entegrasyonuna sahne olan CBS alanında, şimdiye kadar doğrudan ilişkili olmayan teknolojilerin entegrasyonu daha çok görülecektir. Bu kapsamda aşağıdaki gelişmeler daha şimdiden başlamıştır:

- Veri ambarları ile CBS entegrasyonu
- Multi-medya ile CBS entegrasyonu
- Uzman sistemler ve yapay zeka ile CBS entegrasyonu
- Yeni veri toplama teknikleri. Koordinat ve diğer fiziksel büyüklüklerin, örneğin radyasyon ve koordinatın eşzamanlı ölçümü ve doğrudan CBS’ye aktarımı
- Gerçek zamanlı veri toplama, mikrobot algılayıcılar ile sürekli veri toplama

- 3 boyutlu CBS
- Küresel CBS, yani internet üzerinde etkileşimli CBS.” [5]

2.3 CBS İLE VERİDEN ÜRÜN HARİTA ELDE ETME

2.3.1 CBS için Veri Elde Etme Yöntemleri

Coğrafi Bilgi Sistemleri için çeşitli konumsal veri toplama yöntemleri hakkında şu bilgiler verilebilir:

“1) Arazi ölçmeleri: Büyük ölçekli topografik veri, arazi ölçmeleriyle elde edilebilir.

2) Fotogrametrik ölçmeler: Hava fotogrametrisindeki nesne koordinatları bugünkü analog veya dijital stereoplotter’larda saptanabilir ve bilgi sistemlerine doğrudan dahil edilebilir.

3) Uydu verileri: Uzaktan algılama uyduları, dünya yüzeyinden yansıtılan veya yayılan ışınım duyarlı algılayıcılar ile tarayıcılar içeren yapıya sahiptirler. Tarayıcılar, algılayıcıların uçuş hattına dik yollar boyunca tarama yapar veya grid hücrelerinden ard arda ışınım ölçmesi yapma prensibine göre çalışır. Işınım verileri değerlendirilir ve bir harita bu şekilde ortaya çıkarılır. Veri doğruluğu, hem ışınım değerleri hem de geometrik doğrulukların her ikisi için de yapılan düzeltme tekniklerine bağlıdır. Veri bu düzeltmelerden sonra sık sık özel gridleri düzenlemek veya diğer veri grupları ile karşılaştırmak amacıyla tekrar örneklenmek zorundadır.

4) GPS verileri: 24 uyduyu baz alan, bir noktanın 3 boyutlu konumunu birkaç cm doğrulukla belirleyebilen GPS aletleri tarafından sağlanır.

5) Haritaları sayısallaştırmak: El ile sayısallaştırma, konum koordinatlarının ve harita üzerindeki hatlara ait karakteristik noktaların sıra halinde, bir kursor ile kayıt edilmesidir.

6) Haritaları taramak: Tarama işlemi bir faks makinasının çalışma prensibine dayanır: özel konumda özel renklerin optik kayıtları konum ve özellik yani renk veya renk tonu değerleriyle dosyalara dönüştürülür.

7) Varolan sınır dosyalarının kullanımı: Mevcut sınır dosyaları önemlidir. Bunlar topografik dosyalar, navigasyon dosyaları veya istatistiksel veri ile kullanılan sınır dosyalarıdır.

8) Sosyo-ekonomik istatistiksel dosyalar: Uluslararası istatistiksel servisler verilerini harita formunda görselleştirmek veya dosyaları sorgulamak amacıyla CD-ROM'lara yazarak hazırlarlar.

9) Jeo-fiziksel veri dosyaları: Yüzey bilimcilerin sağladığı veri dosyalarıdır.

10) Çevresel veri dosyaları: Çevresel amaçlar için kullanılabilen temel verinin toplandığı dosyalardır.”

2.3.2 Hazırlanan Verilerin Birleştirilmesi

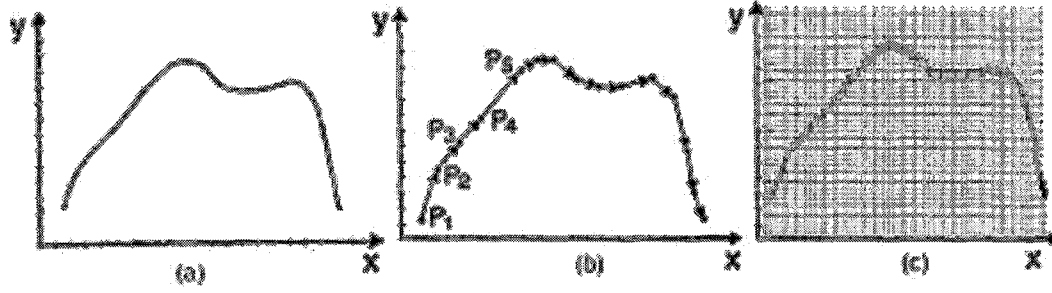
Konumsal veri analizi ve CBS işlemleri için birçok veri kullanılmaktadır. Bu değişik veri dosyalarının doğru birleştirilmesi çok önemlidir. İdeal bir koşulda birleştirilen verilerin tümü benzer veya uyumlu yöntemlere göre aynı tarihte toplanmalı, tanımlanmalı, ölçülmeli ve aynı metot kullanılarak Coğrafi Bilgi Sistemleri'ne dahil edilmelidir. Kullanıcılar ancak dosyaların birleştirilmesinde aynı yöntemleri kullanırlarsa sonuçlarının yeterli doğrulukta olduğundan emin olabilirler.

Pratikte genellikle veri elde etme idealden uzaktır. Veri farklı zamanlarda toplanır, farklı zamanlar için geçerli olabilir, farklı konumsal çözülmeye sahip olabilir ve bazıları varolan haritalardan alınarak bilinmeyen bir mertebede genelleştirilirken, bazıları araziden elde edilir. Bazı dosyalar birkaç düzeltme yöntemi uygulandıktan sonra sisteme girilmiş olabilir. Birkaçı gelişigüzel örneklerle bazıları da eksiksiz yapılan ölçülere dayandırılmış olabilir. Sayısal verinin düzenli örnek noktalarda elde edilmesinin yanında bazı veriler doğrusal olarak bazıları da dizinin geometrik şeklini temel alarak enterpole edilmiş olabilir.

Veri yüzeysel olarak uyumlu hale getirildiğinde analizlerinden çıkarılan sonuçlar doğru olmayacaktır. Analizi yapanlar bu aşamalarda yeterli hassasiyeti göstermeli veya genel Coğrafi Bilgi Sistemi kullanıcıları sonuçları dikkatle yorumlamalıdır.

2.3.3 Verinin CBS Ortamına Aktarımı

Temelde CBS'ye veri aktarılırken, vektör veya raster olarak isimlendirilecek iki veri formatından birisi kullanılır. Şekil 2.9 (a)'da görülen bir çizginin (b)'de vektör ve (c)'de raster formatta gösterimi yer almaktadır.



Şekil 2.9 Bir çizginin vektör ve raster formatlarda gösterimi

Vektör dosyaları, alanlar arasındaki hat veya sınırlar, nokta konumları ve bağlantı hatlarıyla tanımlıdır.

Raster dosyalar ise sınırlar veya ilgili diğer temel değerler ile işlerlik kazandırılmış düzenli gridler üzerinde resim elemanlarının (piksellerin) dizileri olarak tanımlanır.

Vektör dosyalar, haritaların sayısallaştırılması ile üretilir.

Bilgi dijital olarak mevcut olduğunda ölçek daha az önemli olur ve bu yüzden küçültmek ve büyötmek mümkündür. Ölçek, orijinal ölçeğin çok üstüne çıkarılırsa coğrafi elemanlar kötü ve kaba görünecektir ve kesin referanslama için kullanılamazlar. Ancak, CBS ortamında, büyük ve küçük nesneler farklı tabaka ve dosyalarda depolanabilir ve belli bir ölçek değeri geçildiği zaman diğer dosya işlerlik kazanabilir. Örneğin 1/250000'lik ölçeğin ötesine zoom yapıldığında küçük nehirler işlerlik kazanır ve görünür hale gelebilir.

Bir faks makinasında olduğu gibi, bir tarayıcı kullanılırken de siyah-beyaz veya diğer renklerdeki noktalar, çizgiler veya lekelerin optik kayıtları, bilgi ve nitelikleriyle beraber dosyalara dönüştürölür. Düzenli gridler içeren dosyalar, resim elemanlarından (piksellerden) meydana gelir.

Tarayıcının çözünürlüğü yüksekse pikseller küçük olacak ve orijinal resimdeki daha çok detay gösterilecektir. Dünya yüzeyindeki alan ne kadar küçükse, raster resim daha büyük bir doğrulukla orijinal veriyi modelleyecektir.

2.3.4. Verinin Modellenmesi

Gerçek dünyadan veri modeli oluşturulur. Veri yapısı, vektör veya raster formatta olur. Veri yapısı, veriyi nasıl sorgulamak istediğimizle bağlantılı olarak belirlenecektir.

Vektör veri yapısında, veriler nesnelere göre organize edilirler. Verinin geometrik karakteristikleri, koordinat kümeleri ile gösterilir. Bunlar haritada, çizgi olarak - vektörler- birbirlerine bağlıdır. Geometri de niteliğe etiketler ile bağlıdır.

Raster veri yapısında, veri konumsal adres temeline dayanılarak organize edilir. Geometri, grid hücrelerin konumları ile gösterilir. Hücreler adresleri ile niteliklerine bağlanır.

CBS projesinde kullanılacak veri için, hangi veri yapısının uygun olacağına karar verilir.

“Veri yapısını değerlendirirken, kişi, en azından bütünlük, verim, sıralama, çok yönlülük, fonksiyonellik gibi noktaları göz önüne almalıdır.

Bütünlük: Veri yapısının tüm seçilen veriyi gösterebilme kabiliyetidir.

Verim: Seçilen yapının veriye erişimi ne derecede sağladığıdır.

Uygunluk: Verinin toplanma, elde edilme yolu hangi veri yapısının uygun olacağı hakkında ipucu verebilir.

Çok yönlülük: Yapının yeni ortamlara adapte edilebilme olasılığıdır.

Fonksiyonellik: Veri ile uygulanabilir işlemlerin ne olduğu, seçilen yapı ile ne tür sorgulamalara cevap verilebildiği ile ilgilidir.

Gösterilecek verinin doğası, vektör ve raster veri yapıları arasındaki seçimde çok kuvvetli etkindir. Eğer ilgilenilen nesne (örneğin, kablolar, ağlar...vs.) kullanım alanlarında aktif ise, seçimin vektör yaklaşımı olacağı açıktır. Organizasyon uzaktan algılama verisine dayanıyorsa, raster veri yapısı önerilir.” [4]

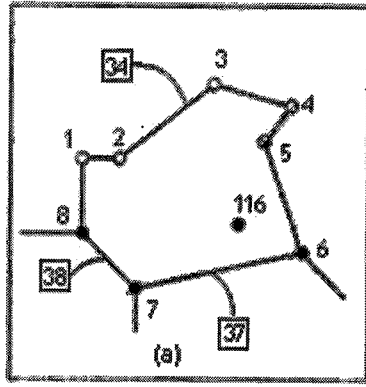
2.3.4.1 Vektör Veri Yapısı Yaklaşımı

Spagetti veri yapısı, vektör yapı türlerinin en basit olanıdır. Tüm nesneler basit parçalar halinde tanımlanır. Şekilde 2.10 (a)’da görüldüğü gibi 6 ve 7 noktaları arasındaki çizgi (X6,Y6) ve (X7,Y7) koordinat kümeleri ile tanımlanır ve 37 etiketine sahiptir. Veri yapısı diğer nesnelerle ilişki kuramaz, mesela kendisine bağlı

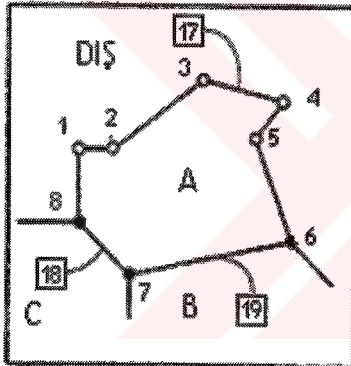
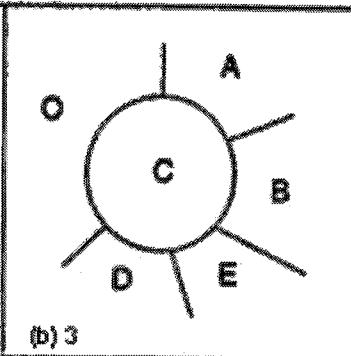
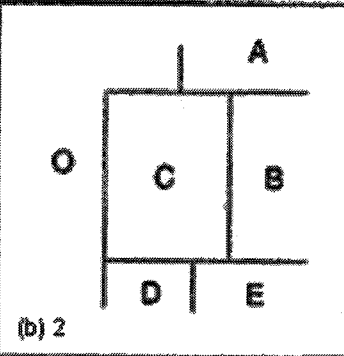
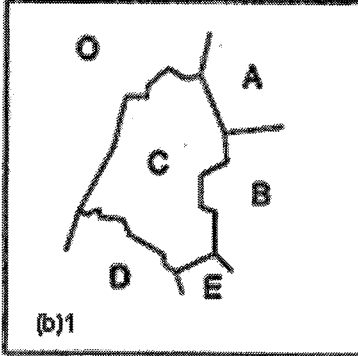
olan, 38 ve 34 çizgilerine dair herhangi bir ilgi içermemektedir. 37 çizgisi ile kuşatılmış alanlara dair de herhangi ilgi bilgisi yoktur. Bu şekil 2.10 (c)'deki birinci haritaya benzerdir ve yapıda tanımlanan nesneler arasında herhangi bir konumsal ilişki göstermemektedir. İçeriği, esas çizime bağlı olmayan sorular cevaplanamamaktadır. Bu bağlantı ancak görsellikle yani görüntü ile gerçekleştirilebilmektedir. Bu yaklaşım 1970'lerin başında tanıtılmıştır.

Daha gelişmiş olan, topoloji içeren vektör veri yapılarıdır. Topoloji, konumsal nesneler hakkında olgunlaşmış ilişkiler tanımlar ve çizgi ve alan nesneleri arasında bağlantıyı kurmak için veya daha karmaşık sorgulamalara cevap bulmak için kullanılır. Topoloji aynı zamanda mümkün genelleştirmelerin en yüksek seviyesi olarak tanımlanabilir. Sonuçta ortaya, şekil 2.10 (b)'de görüldüğü gibi aynı alanın farklı görüntüleri olarak çıkar. Şekil (b)'de C alanının, I' de kendi doğal sınırları vardır. II ve III' de bu sınırlar genelleştirilmiş ancak C alanı ve komşuları ile ilişkileri hala geçerlidir.

Daha kompleks ve gelişmiş olan ise coğrafi ilişkisel veri yapısıdır. Topolojinin yanında, nitelik verisi içeren bir veri sistemine linkler içerir. Bu da daha verimli ve esnek bir yapı sağlar. Bugünün birçok CBS yazılımında bu özellikler bulunabilir. Şekil 2.10 (c) bu yaklaşımın prensiplerini gösteriyor. En basit birim, çizgi parçasıdır. Her parça, başlangıç ve bitiş noktaları bilinmektedir. Bu iki nokta, bu parçaya bir yön vermektedir ve bir sol alan ve bir sağ alan tanımlamayı mümkün kılmaktadır. Şekil (c)'de 17 etiketli çizginin başlangıç noktası, 8 ve bitiş noktası da 6 no'lu bağlantı noktalarıdır. Sol ve sağ alanlar ve sırasıyla dış kısım ve A bölgesidir. Uygulanabilir ise, tüm bağlantı noktaları arasındaki nokta sayısı kaydedilir. Parçaların etiketleri nokta, alan ya da nitelik bilgisi olan tablolara götüren işaretler olarak kullanılabilir. Bunlar veri tabanına link olma fonksiyonu gösterirler. Coğrafi bağlantılı veri yapısı alan toplama, tutarlılık kontrolü ve nitelik verisi bağlamak için daha esnek araştırma işlemleri sağlar.



	kod	koordinat	nitelik
nokta	116	(x, y)	merkez
çizgi	34	$(x_8, y_8), (x_1, y_1), \dots$ (x_3, y_3)	sınır
çizgi	37	$(x_6, y_6), (x_7, y_7)$	sınır
çizgi	38	$(x_7, y_7), (x_8, y_8)$	sınır



(c)

poligon	alan	çevre	nitelik 1	nitelik ...
A	783	1621	...	...

çizgi	uzunluk	baş. nokta	bitiş nokta	sol poligon	sağ poligon	nitelik 1	nitelik ...
17	1032	3	4	DIŞ	A	dış sınır	...
18	211	8	7	A	C	dış sınır	...
19	378	6	7	B	A	dış sınır	...

nokta	koordinat	nitelik 1	nitelik ...
1	x, y	...	...
2	x, y	...	...
3	x, y	...	...
...	...	...	...
8	x, y		

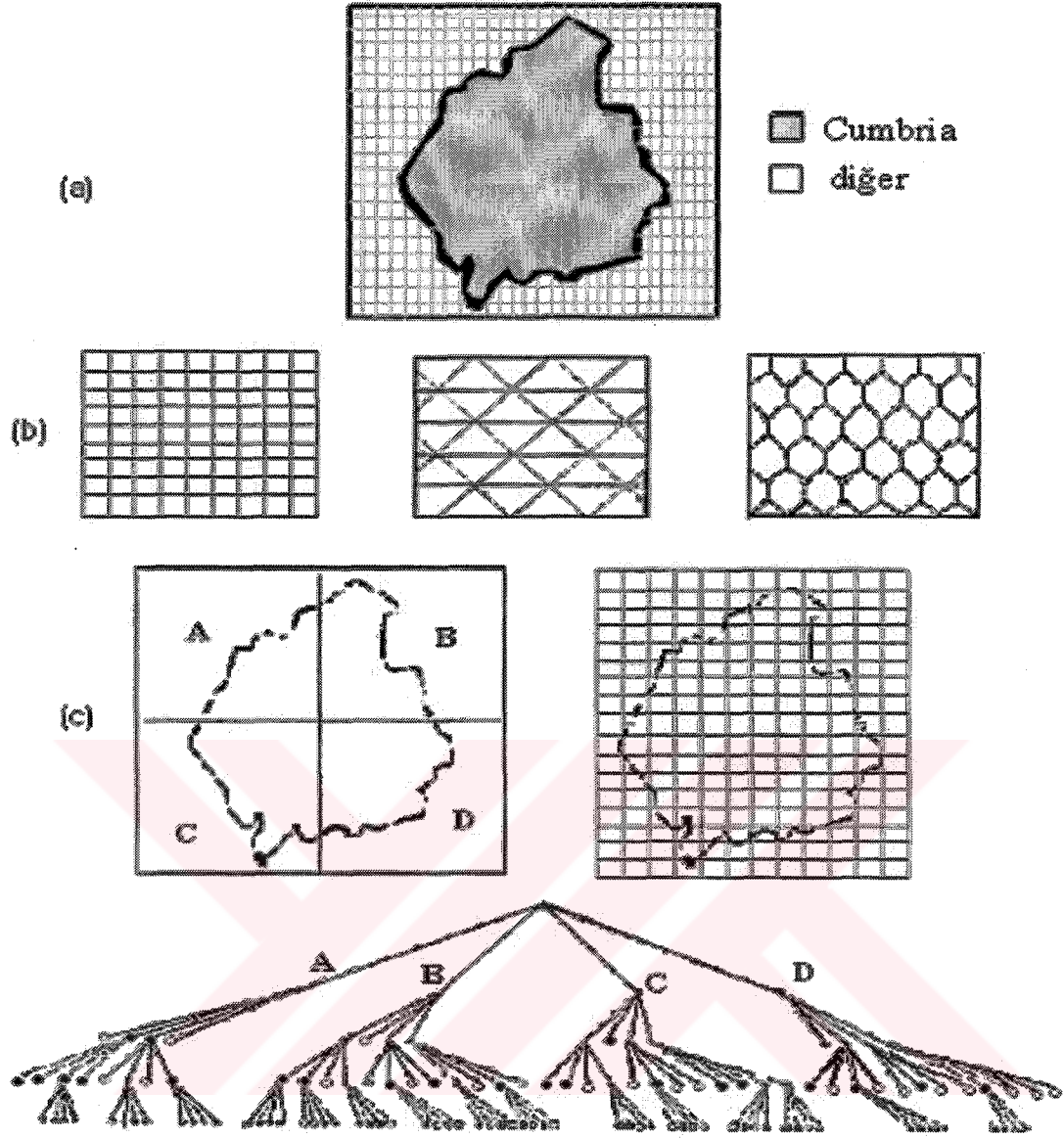
Şekil 2.10 Vektör veri yapısı ve veritabanındaki ifadesi [4]

2.3.4.2 Raster Veri Yapısı Yaklaşımı

Raster yaklaşımında, konumsal birimler, temel coğrafi nesneler yerine referans birimleri olarak etkinlik gösterir. Donanım tarafından en kolay uygulama kareler ile işleme olduğundan kareler en yaygındır. Şekil 2.11 (a)'da basit bir raster yapı içinde

bir coğrafi özelliğin nasıl kaydedildiğini göstermektedir. Gri kareler Cumbria'yı ve beyaz kareler Cumbria alanının dışını tanımlar. Daha çok gri tanıtmak daha çeşitli nesne kaydına olanak sağlar. Ancak her kare için bir nitelikten fazlasını kaydetmek mümkün değildir. Eğer daha fazla niteliğe ihtiyaç varsa, ki bu özellikle GIS ortamında olur, kişi daha fazla raster katman depolamak zorundadır. Karelerin boyutu ki bu çözünürlük olarak adlandırılır, bir raster yapının coğrafi gerçekliği tanımlama şeklidir. Küçük boyut daha iyi bir gösterim verecektir ancak veri kümesi çok yer kaplayacaktır. Bazen veri kaynağı, çözünürlüğü tanımlar. Buna bir örnek, uzaktan algılama teknikleri ile veri toplamak için uyduda bulunan tarayıcının çözünürlüğüdür.

Bir üçgen ya da heksagonun yerine kare kullanmanın bir diğer avantajı, onun aynı şekil ve gösterimin alt birimlerine ayrılabilmesidir. Şekil 2.11 (b)'de gösterilen veri yapısı bu yaklaşıma dayanır. Bu dallanmada, tüm harita bir kare imiş ve dört küçük kareden oluşmuş gibi düşünülür. Kareleri, her kare homojen bir içeriğe sahip olana kadar bölme işlemi sürer. Pratikte ise bu 6 ya da 7 seviyeye kadar uzanır. Şekil 2.11 (c)'deki örnek 4 seviye derinliktedir. Görüntüdeki haritanın dallanması, onun nasıl yapıldığını gösterir. Her koyu daire, Cumbria'yı gösterir, ve her açık daire Cumbria dışındaki alandır. Bir daire bırakacak hiç bir dal olmadığında homojen yani siyah ya da beyaz olduğu düşünülür. Bu temel yaklaşımda bir çok çeşitli varyasyon bulunur.



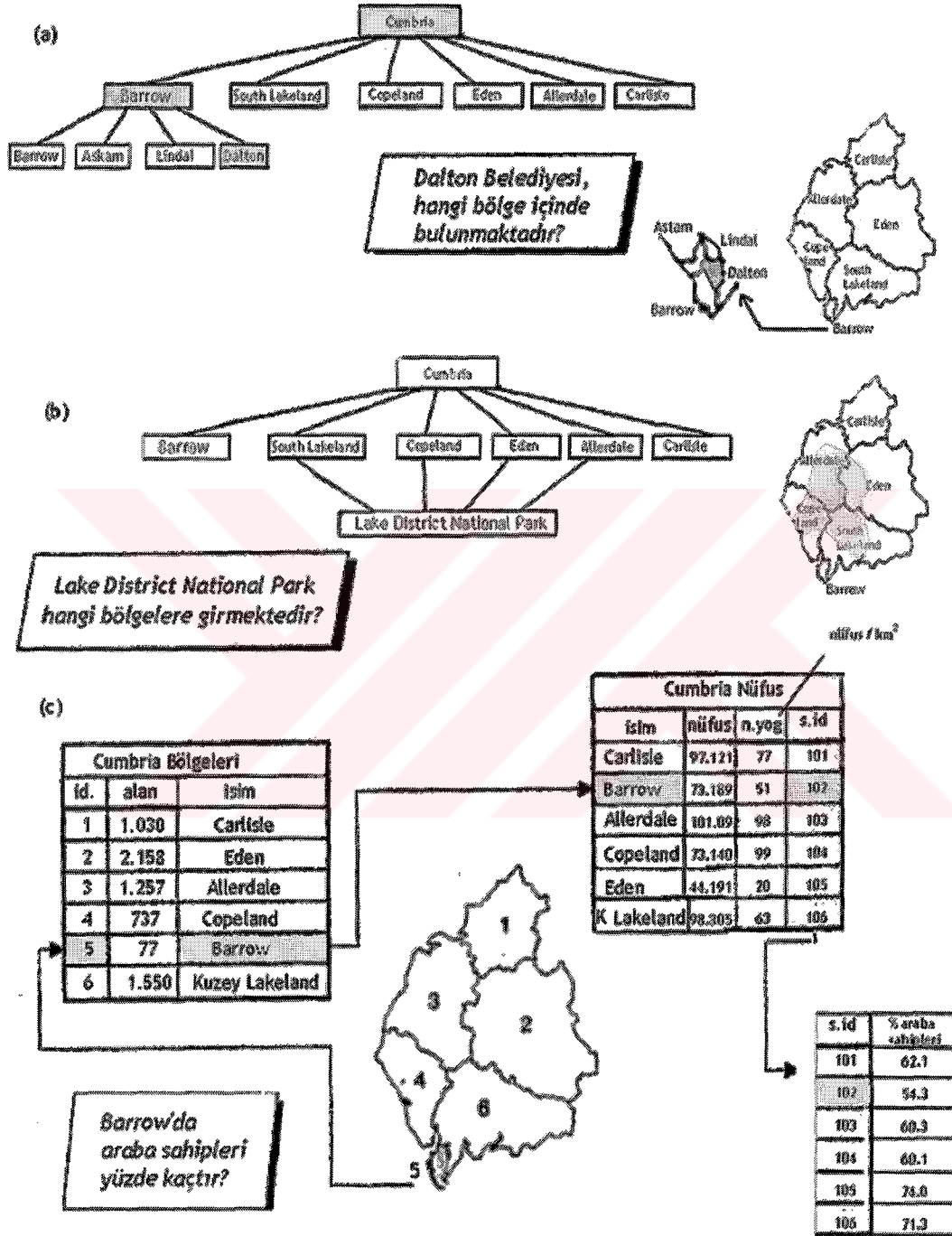
Şekil 2.11 Raster veri yapısını oluşturan hücreler [4]

2.3.4.3 Hem Vektör Hem Raster Yapının Bir Arada Kullanılması

Veritabanı yönetimi sistemlerinde, hem raster hem de vektör veri yapıları kullanılır. Bir veri tabanı yönetim sistemi, şu yapılardan birine sahiptir: Hiyerarşik, ilişkisel ya da ağ yapısı.

Aşağıdaki şekilde, bahsedilen veri yapılarının sorgulamalarda sağlayacakları esneklikler karşılaştırılmıştır. Şekil 2.12 (a)'da sorgu cevabı aranırken sadece belli bir çizgi üzerinde ki bilgilere ulaşılabilmektedir, yani hiyerarşik ilişkiler kurulmuştur. Şekil 2.12 (b)'de ağ yapısı olan bir veritabanının ne tür soruları cevaplayabileceği örnek gösterilerek anlatılmıştır. Şekil 2.12 (c)'de ise bir CBS

projesinde ilişkisel veri tabanı bulunması durumunda ne kadar esnek sorgulama yapılabileceği açıkça görülmektedir.



Şekil 2.12 Farklı veri yapılarının sorgulamalarda sağlayacağı esneklikler [4]

1. Hiyerarşik

Ağaç benzeri bir yapıya sahiptir ve sorular sadece bu ağaç dalları üzerinden sorulabilmektedir ki bunlar şekil 2.12 (a)' de görülebilir. Burada Cumbria her biri küçük adacıklara bölünmüş alanlara ayrılmıştır. Bu sabit yapısından dolayı, sorgulamalarda daha kısa cevap süresine sahiptir. Bu, asıl olarak yönetim becerisine sahip bilgi sistemlerine uygulanmaktadır. Burada soruların doğal yapısı önceden bilinmelidir.

2. Ağ Yapısı

Sorgulama sürecinde daha esnek bir yaklaşıma ihtiyaç duyulduğunda, bir ağ yapısı kullanılabilir. Şekil 2.12 (b) bu yapının dallar boyunca sorgulama ile sınırlanmadığını göstermektedir. Aynı seviyenin veritabanı elemanları, (şekildeki alanlar gibi) sorgulamada birleştirilebilir. Bu veritabanları kullanım şirketlerinde bulunabilir. Ancak hiyerarşik veritabanlarında, sorunun tipi kişi veritabanı yapısını tanımlarken sabitlenir.

3. İlişkisel

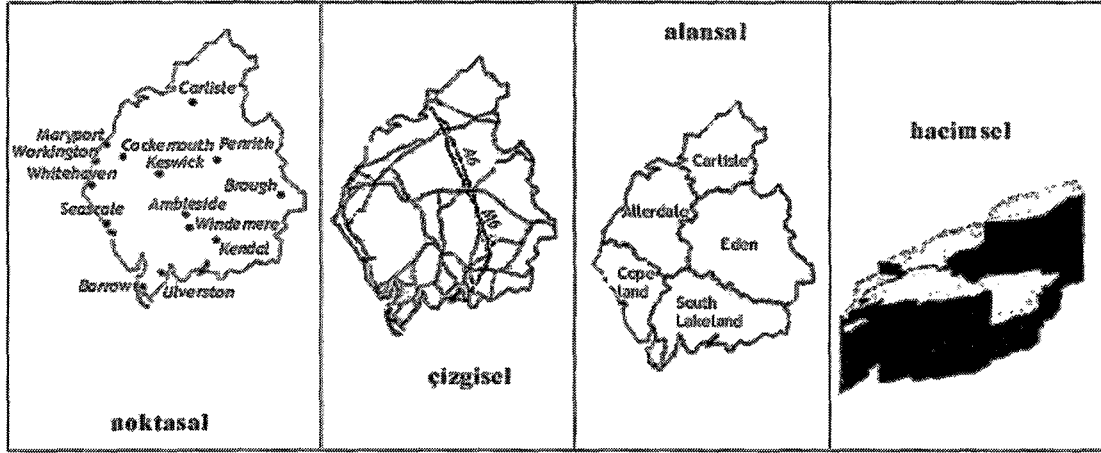
İlişkisel veritabanı yapısı ise daha da esnektir. Her tür sorgulamayı kaldırabilir ve önceden tahmin edilemez ve devamlı değişen sorgulamalara sahip bir ortamda çok kullanışlıdır. Ancak, iki veritabanı yapısından da daha yavaştır. Şekil 2.12 (c) bir örnek göstermektedir. Bir CBS veritabanında Cumbria alanlarının geometrisi bir tabloda saklıdır. İlişkisel veritabanı prensipleri, kişinin bu bilgiye ortak bir değişkeni ulaşılabilir olan başka bir tablodan bağlantısına izin verirler. Şekilde bu, alanın adı ve onun istatistiksel belirleyicisidir. Bu farklı tablolar arasındaki bağlantılar kişinin daha karmaşık sorular sorabilmesine olanak tanır.

2.3.5 CBS Verisinin Görselleştirilmesi

CBS verisi gösteriminde, veri sınıflandırması önemlidir. Veriler farklı özelliklerine göre sınıflandırılabilir. Bu sınıflandırmalar, veri gösterimi tasarlanırken verinin yapısını anlama bakımından yardımcı olacaktır.

1. Noktasal, çizgisel, alansal, hacimsel gösterimler:

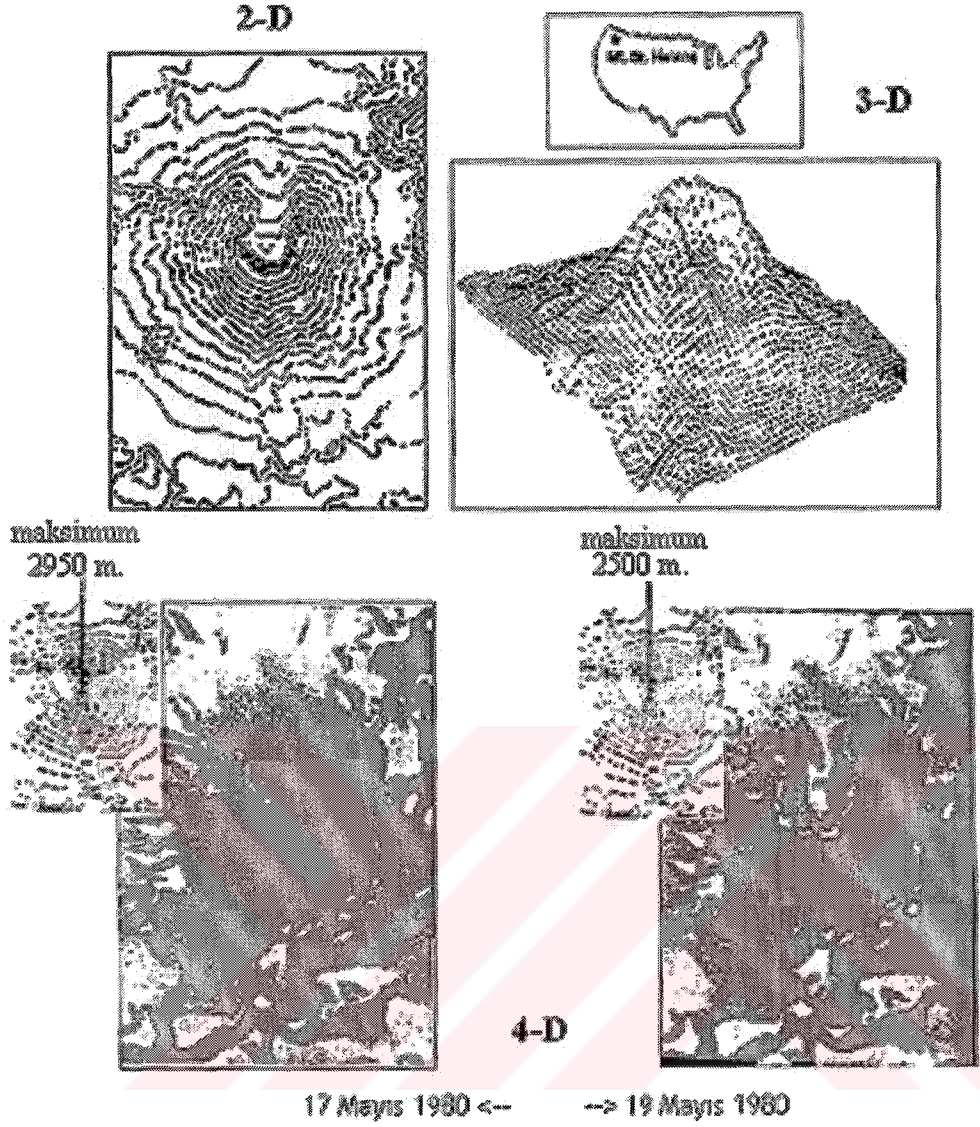
Coğrafi nesnelerin gösteriminde noktasal, çizgisel, alansal ve hacimsel gösterimler kullanılabilir. Şekil 2.13'de, bu tür gösterimler yer almaktadır.



Şekil 2.13 Coğrafi nesnelerin noktasal, çizgisel, alansal ve hacimsel gösterimi [4]

2. 2, 3, 4 Boyutlu gösterimler:

Coğrafi nesnelerin farklı boyutlandırmalarla gösterimi yapılabilir. Şekil 2.14'te 2, 3 ve 4 boyutlu gösterimlere örnekler verilmiştir.



Şekil 2.14 Coğrafi nesnelerin 2, 3 ve 4 boyutlu gösterimleri [4]

Eğer uzaklık söz konusu ise, konumsal veri düz bir çizgi şeklinde gösterilebilir. Noktalar, çizgiler ve dokularla iki boyutlu gösterim planimetrik bir haritaya karşılık gelir. St.Helens dağını gösteren yukarıdaki eş yükselti haritası, bunun için iyi bir örnektir. Üç boyutlu bir gösterim elde edebilmek için ya kartondan St.Helens dağını fiziksel modelini oluşturmalı ya da monitör ekranında bütün yüzlerinin görülebilecek şekilde döndürülmesi sağlanmalı veya kırmızı ve yeşil camlar kullanarak 3 boyutlu görüntü sağlamak tercih edilebilir.

Perspektif ve bir gizli çizgi algoritması kullanılarak modelin çizilmesi, şekilde gösterildiği gibi üç boyutlu bir görünüm izlenimi vermektedir. Bu tanımlama 2,5 boyutlu bir gösterim olarak da bilinir. Tepe gölgelendirmeli haritalar buna başka bir

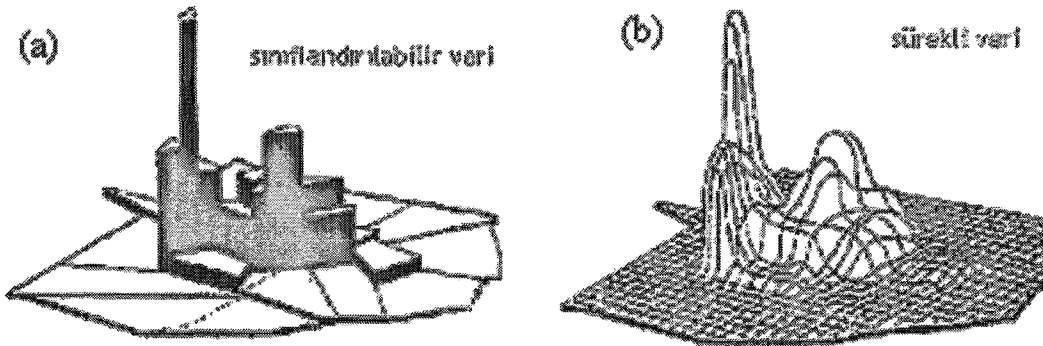
örnek teşkil ederler. Eğer zaman boyutu da eklenecek olursa gösterim dört boyutlu bir hale gelir. Örneğin bir nesnenin iki farklı zamanda yan yana sunulması sonucunda 17 Mayıs ve 19 Mayıs 1980'deki görüntülerin geometrik ve niteliksel değişikliklerini araştırmak mümkündür. [4]

3. Harita üzerinde dokunulabilir ve dokunulamaz nesneler:

Coğrafi nesne yer yüzeyinin üstünde, üzerinde veya altında belli bir konumu ve biçimi olan soyut veya somut herhangi bir varlık veya olgu olabilir. Köprüler, yollar, ormanlık alanlar somut (dokunulabilir) coğrafi nesnelere birer örnektir. Diğer taraftan idari sınırlar, havayolu güzergahları, sorumluluk bölgeleri, şehirlerdeki gürültü alanları ise soyut (dokunulamaz) tipte coğrafi nesnelerdir. [5]

4. Coğrafi nesnelerin sınıflandırılabilir veya sürekli oluşları:

Gerçek dünyaya ait nesneler ve fenomenlerin sınıflandırılabilir veya sürekli oluşları, haritalama işlemlerinde önem taşımaktadır. Sınıflandırılabilir nesneler, ev ve akarsu gibi dokunulabilir veya devletler, numaralandırılmış alanlar gibi dokunulamaz nesneler olabilir. Sürekli gösterimler ise yağış miktarı verileri, gravite alanı verileri gibi dokunulabilir veya ölçülebilir nesnelerin yanı sıra, eş yağış eğrileri gibi modeller üzerine temellendirilmiş de yapıda da olabilirler.” [4] Şekil 2.15’de (a) sınıflandırılabilir veri gösterimi, (b) sürekli veri gösterimi yer almaktadır.



Şekil 2.15 Sınıflandırılabilir ve sürekli veri gösterimi [4]

5. Niteliksel veya niceliksel verinin ölçüm yapılarak sınıflandırılması :

Görselleştirme işlemleri niteliksel bilginin doğasını anlayabilmek için gereklidir. Bu nitelikler görünür karakterler (çeşitli ağaçlar) olabileceği gibi görünmez karakterler

de (sıcaklık) olabilmektedir. Veriyi niteliksel veya niceliksel olsun, ölçüm yaparak tanımlayabiliriz. Aşağıda ölçüm çeşitleri ve farklılıklarından bahsedilmiştir:

Nominal ölçüm: Biri diğerinden daha önemli olmaksızın farklı olan niteliksel değerler için.

Dereceli ölçüm: Niteliksel değerler birbirinden farklıdır, bazıları diğerlerine göre daha önemli olduğundan düzenleme yapılabilir.

Aralıklı ölçüm: Niteliksel değerler birbirinden farklı olup, düzenlenebilirler ve belli aralıklara ayrılarak sınıflandırılabilirler. Sıcaklık bunun için iyi bir örnektir. Sıfır noktası tesadüfi olarak belirlenmiştir, bu nedenle 64 °F, 32 °F'nin iki katıdır demek olanaksızdır ve bu değerleri Celcius'a çevirecek olursak 18°C ve 0 °C haline dönüşür.

Oranlı ölçüm: Konunun yorumu için gerek duyulan birden fazla değişken varsa, dikkate alınan ölçmeler birbirleriyle ilişkilendirilebilir. [4]

2.3.6 Görselleştirilen Veri ile Çalışmak

Nesneler ve birbirleriyle ilişkilerinin kartografik kurallar kullanılarak görselleştirilmesi ile harita ortaya çıkar. Veriler artık bilgisayar ortamına atılmış, üzerinde çalışılabilir, analiz yapılabilir ve tanıtılabilir hale gelmiştir. Harita nokta, çizgi, alan ve dokulardan, semboller ve işaretlerden meydana gelmiştir. Bunlar şekil, büyüklük, doku, renk, gri renk tonu ve yön açısından gösterdikleri çeşitlilik ile kullanıcıya bazı bilgileri aktarır, bir anlam ifade ederler. Çünkü her işaret, gösterim, temsil ettiği nesne ve bu nesnenin nitelikleri ve ilişkileri ile bağlantılıdır ve harita kullanıcısına konumsal ilişkileri yansıtmaktadır.

Haritalardaki görselleştirmeler sayesinde, nesnelerin konumsal ilişkileri görünür hale gelir. Geometrik ve niteliksel verinin kombine edilmesiyle, insan farklı nesne kategorilerinden elemanların konumlarının birbirini nasıl etkilediğini idrak etmektedir. Öncelikli ilişkilerin yanı sıra ikinci derece ilişkileri de anlamak mümkün olur. Örneğin nesnelerin çizgisel veya alansal referans birimleriyle ilişkileri gibi. Bunlar alan-nüfus ilişkisi, araba sayısı-otayol ağı uzunluğu ilişkisi, bahçivanlık-tarım alanları ilişkisi olabilmektedir. Zaman gibi üçüncü veyahut daha yüksek dereceden ilişkiler de ortaya çıkarılabilir.

2.3.7 CBS Ürünü: Harita

Konumsal verinin geometrik, niteliksel ve zamansal olmak üzere üç bileşeni olduğu belirtilmişti. Bu 3 bileşen sorgulanarak haritalar aracılığı ile yapılan konumsal, tematik ve zamansal karşılaştırmalar, CBS teknolojisi ile çok daha kolay ve görsel açıdan da tatmin edici olmuştur.

Konumsal karşılaştırma yapmak, aynı ölçekte farklı yerlere bakarak, bazı özelliklerin bağlantılı veya farklı olduğunu görmek demektir. Tematik karşılaştırma yapmak, aynı alana ait farklı temaları gösteren haritalara bakarak, temanın konumsal dağılımının benzer veya farklı olduğunu görmek demektir. Zamansal karşılaştırma yapmak, aynı alanın farklı zamanlardaki görüntülerine bakarak karşılaştırmak demektir.

2.3.7.1 Konumsal Verinin Geometrik Bileşenini Karşılaştırmak

Tek bir tema üzerinde yoğunlaşarak, bu temanın iki farklı alandaki durumunu karşılaştırmak harita üzerinde çok kolaydır.

Hassas bir sonuç için, karşılaştırılan haritaların aynı metotlarla toplanıp düzenlenmeleri, aynı ölçekte olmaları ve aynı seviyede genelleştirilmiş ya da aynı sınıflandırma metotları ile sınıflandırılmış olmaları gerekir.

2.3.7.2 Konumsal Verinin Niteliksel Bileşenini Karşılaştırmak

Eğer belirli bir alana ait, iki veya daha çok tema, aynı metoda dayanarak haritalanıyorsa, bu haritaları karşılaştırarak benzerlik ve farklılıkları değerlendirmek mümkündür.

2.3.7.3 Konumsal Verinin Zamansal Bileşenini Karşılaştırmak

Artık kullanıcılar, anlık belirlenmiş veri analizleriyle yetinmemekte ve bir süreci analiz etmek istemektedirler. Bu tip bir analizde sık karşılaşılan amaç, konum üzerinde zamanla değişen tipik yapıları belirlemektir. Daha sonra değişim tek bir harita ile gösterilebilir.

Buna verilebilecek bir örnek, nüfus merkezinin kaymasını gösteren haritalardır. Nüfus merkezi, her nüfus sayımı için tüm sayımların merkez koordinatlarının belirlenmesi ve toplanması hesabı ile yapılır.

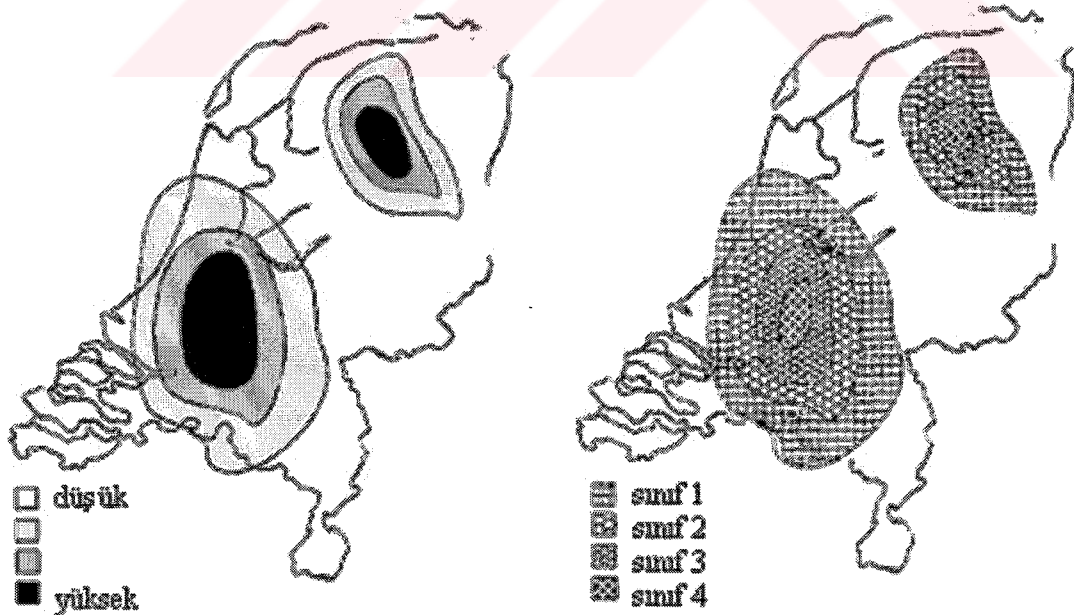
Bir alternatif de, her bir zamanda ayrı bir anı gösteren basit harita serilerinin kullanımıdır. Bu haritaların kıyaslanması, kullanıcıya değişimle ilgili fikir verecektir.

Daha yeni olan diğer bir alternatif de dinamik görüntülerin veya animasyonların kullanılmasıdır. Televizyonda, günlük hava raporlarını sunmak üzere meteorolojik değişimleri gösteren animasyonların kullanılması örnek olarak verilebilir.

2.3.7.4 Ürün Haritada Sonucun Kullanıcıya Hissettirilmesi

Harita ile sağlanmaya çalışılan, harita kullanıcısının gözünde dünya yüzeyini canlandırabilmektir. Harita ile karşı karşıya kalan kişinin, gerçek dünya ile zihinsel düşüncelerini eşleştirmesi arzu edilir.

Aslında, gerçek dünya ve modeli arasında benzerlik bulunmaktadır, fakat hiçbir zaman tamamen modelleme yapılamaz. Verilerin toplanması veya ölçülmesindeki zaman sarkmaları, sunulmak istenenin dışındaki veri katmanlarının gösterilmemesi, genelleştirmeler, göz ardı edilebilecek verilerin atlanması her modellemede yaşanacaktır. Kullanıcının karşısına gelen, sadece doğal olayın algılanması için sunulan bir modeldir. Harita ile amaçlanan da çoğu zaman düz bir gösterimden çok farklı olarak bir sonuca vardırması, fikir uyandıran iletişim aracı olarak kullanılmasıdır.



Şekil 2.16 Yağış miktarları dağılımının iki farklı yöntemle gösterimi [4]

Şekil 2.16’da, sağ tarafta yağış miktarının sınıflarının dağılımı görselleştirilmiştir. Kolaylıkla her sınıf dağılımı görülmektedir, fakat farklı yağış dağılım sınıfları olduğu fikrini oluşturmaktadır. Bu doğa olayı dağılımında yansıtılmak istenen fikir, düzenlenen ile aynı değildir. Sol taraftaki gösterim ise konunun doğasına daha uygun bir gösterimdir.

“Harita yapılırken iletişim amacına uygun düzenlenmelidir, çünkü doğal olay bir bütün gibi şekillendirilmelidir.” [4]

Haritaların kullanımı ve yapım bilgileri yöntemi ile haritadaki bilgi en iyi yolla sunulduğunda, harita kullanıcıları tarafından, bilginin kontrol edilebilmesi mümkün olacaktır.

Özet olarak, CBS ürünü haritalar doğal olayın sunumunu sağlar, coğrafi verilerin geometrik niteliklerini, öznitelikleri ve konum bilgilerini bulundurur. Konum x ve y koordinatları kullanılarak tanımlanır. Özniteliklerin sunumunu göze çarptırıp yoruma elverişli yapmak, sunumun asıl amacıdır.

Sıklıkla sonuç haritası, üreticinin analiz boyunca temin ettiği her bilgiyi içermez. Doğal olayın gösterilmesi için genelleştirme veya sınıflandırma yapılmış olabilir. Kullanıcının haritadan çıkardığı bilgi, orijinal bilgiyle % 100 üst üste örtüşmez, ancak zaten esas ulaşılmak istenen, doğru genellemeler ve sınıflandırmalarla sonuca götüren harita üretebilmektir.

3. VERİTABANI SİSTEMLERİ

3.1 VERİTABANLARI

3.1.1 Veritabanı

Veritabanı, bir veri topluluğudur. Bu veri topluluğu, bir amaca yönelik olarak bir araya getirilmiş verilerden oluşur. Ancak, veritabanı sadece bir veri yığını değildir, verilerin kendi aralarındaki ilişkileri de saklayan düzenlenmiş bir veri kümesidir. Bu sistem, bilgisayar ortamında kayıt tutan bir sistemdir. Veriler kayıt edilir ve bakımı yapılır. Veritabanı kullanımı ile bir organizasyonda verilerin merkezi kontrolü sağlanır. Veritabanlarının amacı büyük miktardaki kurumsal verileri işlemektir. Veriler düzenli olarak elektronik ortamda kaydedilir düzenli olarak yedeklenen ve kontrol edilen bu veriler çok sayıda uygulamanın ve kullanıcının hizmetine sunulur. Veritabanı sistemleri değişik şekillerde olabilir. Kapasite, hız, güvenlik, veri transferi, uyumluluk, işletim sistemi, kullanıcı ile ilişki gibi daha bir çok alanda farklılıklar gösteren veritabanı uygulamaları vardır. Buna rağmen, hepsinin ortak özelliği ve veritabanı kullanımının amacı, “büyük miktarlardaki verilerin hızlı ve güvenli bir şekilde gereksinim duyulan bilgiye dönüştürülmesi” olarak özetlenebilir. [9]

Bir veritabanı sistemi, birçok kullanıcının paylaşımına açık olan geniş bir veri kümesidir. Bu sistem, veri depolanması, düzenlenmesi, sorgulanması işlemlerini sağlar. Bu işlemleri gerçekleştirme mekanizmasına veri yönetimi adı verilir.

Bir veritabanı sistemi depolanan bilginin güvenliğini sağlamalıdır. Bu güvenlik hem sistemde meydana gelebilecek aksaklık ve bozulmalara hem de yetkisiz erişime karşı olmalıdır. Özellikle veri çok sayıda kişi tarafından kullanılacaksa olası yetkisiz ve kuraldışı kullanımlar ve bunların sonuçlarına karşı sistem güvenli olmalıdır.

Birçok işletmede verinin önemi, onun verimli yönetilmesi için gerekli tekniklerin ve buna ilişkin çok sayıda kavramın geliştirilmesine yol açmıştır.

3.1.2 Veritabanı Sistemlerinin Gelişimi

3.1.2.1 Klasik Dosya Sistemleri

Veritabanı sistemleri geliştirilmeden önce, bilgiyi bilgisayarda saklamanın yollarından biri ayrıık dosyalar içinde depolamaktı. Bu tür bir yapıda sistem, bilginin kullanıcılarca erişilmesi, kullanılması ve işlenmesi olanağı sağlayan pek çok uygulama programı içerir. Bu uygulama programları, sistem programcıları tarafından organizasyonun ihtiyaçlarına göre yazılır. Yeni ihtiyaçlara gereksinim duyuldukça yeni uygulama programları sisteme eklenir. Bu da zamanla, daha çok dosya ve daha çok uygulama programının sisteme eklenmesi anlamına gelir.

Veritabanı yönetim sistemleri geliştirilmeden önce işletmeler bilgilerini bu tür programlarda saklıyorlardı. İşletme bilgilerini bu tür sistemlerde saklamanın pek çok dezavantajı vardır.

3.1.2.2 Klasik Dosya Sistemlerinin Dezavantajları

Bu dezavantajlar aşağıdaki gibi özetlenebilir:

- Veri tekrarı ve veri uyumsuzluğu:

Uzun bir süreç içinde uygulama programları ve oluşturulan dosyalar, farklı programcılar tarafından hazırlandığından bu dosyaların farklı formatlarda olması ve programların da çeşitli programlama dillerinde yazılmış olmaları olasıdır. Bu farklı zamanlarda yazılmış veriler arası uyumsuzluk ortaya çıkaracaktır.

Ayrıca, aynı bilginin pek çok yerde kopyası olması gerekebilir. Bu tekrarlama daha fazla disk alanı ve daha yüksek maliyete yol açar. Aynı zamanda, bu tür bir sistemde veri uyumsuzluğu da meydana gelebilir. Çünkü herhangi bir değişiklik ve güncellenmenin, eksiksiz olarak verinin olduğu tüm kopyalarda yenilenmesi gerekir.

- Veri erişiminde güçlük:

Veritabanlarında pek çok veri bulunmaktadır. Bunların arasından istenen özelliklere uyan kayıtların listesi çıkarılmak istendiğinde iki seçenekten biri tercih edilir. Birinci

yol, tek tek tüm kayıtları taramak ve manuel olarak bu işlemi gerçekleştirmektir ki bu tatmin edici bir yol değildir. İkinci olarak da bir programcıya bu işlemi gerçekleştirecek bir program yazdırmaktır ki bu da geçici bir çözümdür. Manuel işlem karmaşık tür sorgulamalarda çok zorlaşacağı gibi, programcılar da farklı diller kullanmak isteyeceklerdir. Zamanla farklı bir ihtiyaç geliştiğinde, yine tatmin edici olmayan veya geçici bir çözüm sunan iki seçenek karşımıza çıkacaktır. Bu durum, daha kullanışlı bir sistemin geliştirilmesi gerektiğini açıkça göstermektedir.

- Veri İzolasyonu:

Veriler, farklı dosyalarda tutulduğundan dolayı, farklı veri formatlarında olabilirler ve yeni uygulama programları ile eski usulde depolanmış veriler üzerinde işlem yapabilmek zordur.

- Entegrasyon Sorunu:

Veritabanında depolanan bazı veri değerleri için alt ve üst sınırlar verilebilir veya bir değerin, belli bir miktardan az olamayacağı gibi bir kısıt olabilir. Programcılar bazı uygulama programlarına, bu kısıtlara uyulmasını sağlamak üzere gereken kodları ekleyeceklerdir. Ancak, bu kısıtlar değiştirildiğinde, ya da başka kısıtların da göz önüne alınması gerektiğinde, bunların uygulama programlarında değiştirilip sisteme entegre edilmesinde sorunlar çıkmaktadır.

- Atomicity Problemi:

Tüm diğer mekanik ve elektronik aletlerde de olabileceği gibi, bilgisayar sistemleri, başarısız veya geçersiz bir işlem yürütebilir, kilitlenebilir, veya bozulabilir. Pek çok uygulamada, bir hata oluştuğunda ve farkedildiğinde, verilerin hata oluşmadan önceki durumlarında olup olmadıklarından emin olabilmek hayati önem taşır. Örneğin bir bankacılık işleminde A hesabından B hesabına para aktarılırken bir hata meydana gelmiş olsun. Bunun sonucunda A hesabından para çıkmış ancak B hesabına yatmamış ya da A hesabından para çıkmamış ancak B hesabına da yatırılmış gözükabilir. Sistem böyle bir yanlışlığa imkan vermemelidir. Yani ya tüm işlem yapılacak ya da hiçbir işlem yapılmayacak şekilde tasarlanmalıdır. Bunun klasik dosya sistemleri ile gerçekleştirilmesi ise zordur. [10]

- Eş zamanlı erişim sorunları:

Tüm sistem kullanıma hazırlanıp, hızlı bir yanıt alma süresi de sağlanabildikten sonra pek çok sistem kullanıcıların verilere eş zamanlı erişimine ve eş zamanlı veri girişine izin verir. Bu çok kullanıcıli ortam ve eş zamanlı veri girişi, veri tutarsızlığına sebep olmamalıdır.

- Güvenlik problemleri:

Veritabanı sistemi kullanıcılarının tümünün, kayıtlı tüm veriye erişebilmeleri istenmez. Kullanıcı gruplarını ilgilendiren veriler bellidir. Bazı kullanıcıların detaylara veya sistem bilgilerine erişmesi gereksiz veya tehlikeli olacaktır. Bazı kullanıcıların ise bu tür veriye erişim yetkileri bulunmalıdır. Sistem, yetkilendirme konusunda tatmin edici olmalıdır.

Tüm bu zorluklar veritabanı yönetim sistemlerinin gelişimini hazırlamıştır.

3.1.2.3 Veritabanı Sistemlerinin Avantajları

Veritabanı sistemlerinin pek çok avantajı olduğu açıktır. Veritabanı sistemlerinde veri tekrarları ya tamamen ortadan kaldırılır yada en aza indirilir. Veritabanındaki tekrarlı verilerin, veritabanının her yerinde hep aynı şekilde tutulması sağlanarak, tutarsızlıkların oluşması önlenir. Veri tekrarları en aza indirileceğinden, disk alanı israfı engellenir. Veritabanı sistemlerinde standart bir sorgu dili kullanılabilir. “Özellikle SQL dilinin farklılıklarının giderilmesi, standardizasyonunun sağlanması için ANSI (American Standards National Institute) ve ISO (International Standards Organization) bir araya gelerek bir çalışma başlatmış ve 1987 yılında resmi olmayan ilk SQL standardı olan SQL1 ortaya çıkmıştır. Daha sonraki yıllarda gelişen ihtiyaçlar ile yeni SQL standartları da yayınlanmıştır.” [9]

Bunun yanında veritabanı sistemlerinin getirdiği riskler de bulunmaktadır.

Veritabanı sistemlerinin en önemli dezavantajı maliyetleridir. Veritabanı yönetim sistemi paketleri çok pahalı oldukları gibi bunların kurulumu ve bakımlarının da maliyetleri yüksektir. Bir diğer dezavantaj, veritabanı sistemlerinin baştan hatasız olarak tasarlanmasının gerekmesidir. Genellikle konusunda uzman bir veritabanı tasarımcısına ihtiyaç duyulur.

3.1.3 Veritabanı Yönetim Sistemleri (DBMS)

Bir veritabanının kurulması, üzerinde çeşitli işlemler yapılması için gerekli bilgisayar yazılım sistemine veritabanı yönetim sistemi (Database Management System, DBMS) denir. Bir veritabanı yönetim sistemi, hem kendi arasında ilişkili bir veri topluluğunu, hem de bu veriye erişmek amacıyla kullanılacak programları içerir. DBMS'nin asıl amacı, veritabanı bilgisini depolamaya ve işletmeye elverişli ve verimli bir ortam sağlamaktır. Veritabanı Yönetim Sistemi standart bir dil ile uygulamalara ve kullanıcılara hizmet verir.

3.1.3.1 Veritabanı Yönetim Sistemleri Bileşenleri

Bir veritabanı yönetim sisteminde şu bileşenler bulunmaktadır.

- Veri erişimini ve veri işlemeyi sağlayan arabirim : Verileri silme, değiştirme, ekleme, düzenleme işlemlerinin yapılabileceği bir arabirim.
- Uygulama ve geliştirme olanağı sağlayan arabirim : Uygulama geliştiriciler için uygun ortam sağlar.
- Veri modeli : Veritabanı tasarımı için geliştirilmiş bir arabirim.
- Veritabanı Dili: Veritabanı dili ise uygulama geliştiricilerin, veritabanı tarafında kullanacakları dildir.
- Sorgulama Dili: Sorgulama için en yaygın olarak SQL adı verilen yapısal sorgulama dili kullanılır. Farklı veritabanı sistemleri için farklılıklar gösterir, ancak SQL temelde standart bir dildir.
- Veri sözlüğü : Veri yapısını gösterir. Sistem kataloğunda bulunur.
- Güvenlik sistemi : Kullanıcıların izin ve yetkilerinin düzenleneceği ortam.
- Raporlama : Sorgu sonuçlarının rapor halinde üretimini sağlayan araçlar.

3.1.3.2 Veritabanı Yönetim Sistemlerinin Fonksiyonları

Bir veritabanı yönetim sisteminin (VTYS) fonksiyonları:

- Veritabanını oluşturmak
- Veritabanını yönetmek
- Veri saklamak
- Veri erişimi
- Veri güncelleme
- Kullanım kolaylığı: Veritabanı bilgisayar konusunda uzman olmayan kişilerin de rahatlıkla kullanacakları bir yapıda olmalıdır.
- Bir grup işlemi yapabilme : Veritabanında bir grup işlem tam olarak yapılabilmelidir. DBMS, mantıksal işlem gruplarının kullanılmasını sağlamalıdır. İşlem grupları bir grup işlemin tek bir blok olarak ele alınmasını ve işlenmesini sağlar. İşlemlerden birinin yapılamaması durumunda tüm işlem grubu fonksiyonu iptal edilir. Bir işlem grubundaki işlemlerin ya hepsi yapılır ya da hiçbiri yapılmaz.
- Aynı anda işlem yapabilme : Çok kullanıcının paylaşımına açık olan veritabanları, bu kullanıcıların aynı anda veritabanını sorunsuz olarak kullanmasına olanak vermelidir.
- Yedekleme, kurtarma, bakım hizmetleri : DBMS, verilerin yedeğinin alınması bozulan bilgilerin düzeltilmesi gibi bazı konularda hayati önemi olan hizmetleri vermelidir.
- Yetkilendirme : DBMS, farklı kullanıcı gruplarının farklı verilere erişimine izin verecek bir yetkilendirme sistemi düzenleyebilmelidir. Ayrıca, verilerin yetkisiz kişiler tarafından kullanılmasını önlemelidir. Kullanıcılar bir kontrol sisteminden geçerek sisteme erişebilmelidir. Bu, bir kullanıcı adı ve bir parola denetimi yapan kontrol sistemidir.

- İletişim : Veritabanındaki verilerin diğer programlarca girdi olarak kullanılması istenebilir, bu tür bir çalışma için DBMS'nin bu programlarla iletişim kurabilmesi ve verileri gönderebilmesi çok önemlidir.

- Bütünlük : Veritabanındaki veriler, aralarındaki ilişkiler ile korunmalıdır.

- Program-Veri bağımsızlığı : Klasik dosya sistemlerinde olduğunun tersine, dosyaya erişen programların yapısı, dosyadaki verilerin yapısından bağımsız olduğundan, yapıda meydana gelen değişiklikler programlarda da değişiklik gerektirmez.. Erişim programları, herhangi bir özel dosyadan bağımsız olacak şekilde yazılmışlardır. Veri dosyalarının yapısı, DBMS kataloğunda erişim programlarından ayrı olarak saklanır. Veri dosyalarının yapısında meydana gelen bir değişiklik, sadece sistem kataloğunda değişikliğe sebep olur. Uygulama programları bundan etkilenmez. [9-11]

3.1.3.3 Veritabanı Yöneticileri

DBMS kullanımının en önemli sebeplerinden biri de hem verinin, hem de erişim sağlayan programların merkezi kontrolünü sağlamasıdır. Tüm sistem üzerindeki merkezi kontrolü sağlayan kişi 'veritabanı yöneticisi' dir. Kısaca DBA (Database Administrator) ile de gösterilen veritabanı yöneticisinin sistem üzerindeki fonksiyonları şunları içerir:

- Şema tanımı: Veritabanı yöneticisi, orijinal veritabanı şemasını, pek çok tanımlama kümeleri yazarak oluşturur. Bu tanımlamalar, DDL (Data Definition Language) derleyicisi tarafından data dictionary içinde saklanacak tablolar haline getirilir ve veritabanı oluşturulmuş olur.

- Depolama yapısı ve erişim metotlarının tanımı: DBA uygun depolama yapısını ve erişim metotlarını, tanımlamalar kümesi halinde yazar ve bunlar veri depolama ve veri tanımlama dili DDL derleyicisi tarafından oluşturulur.

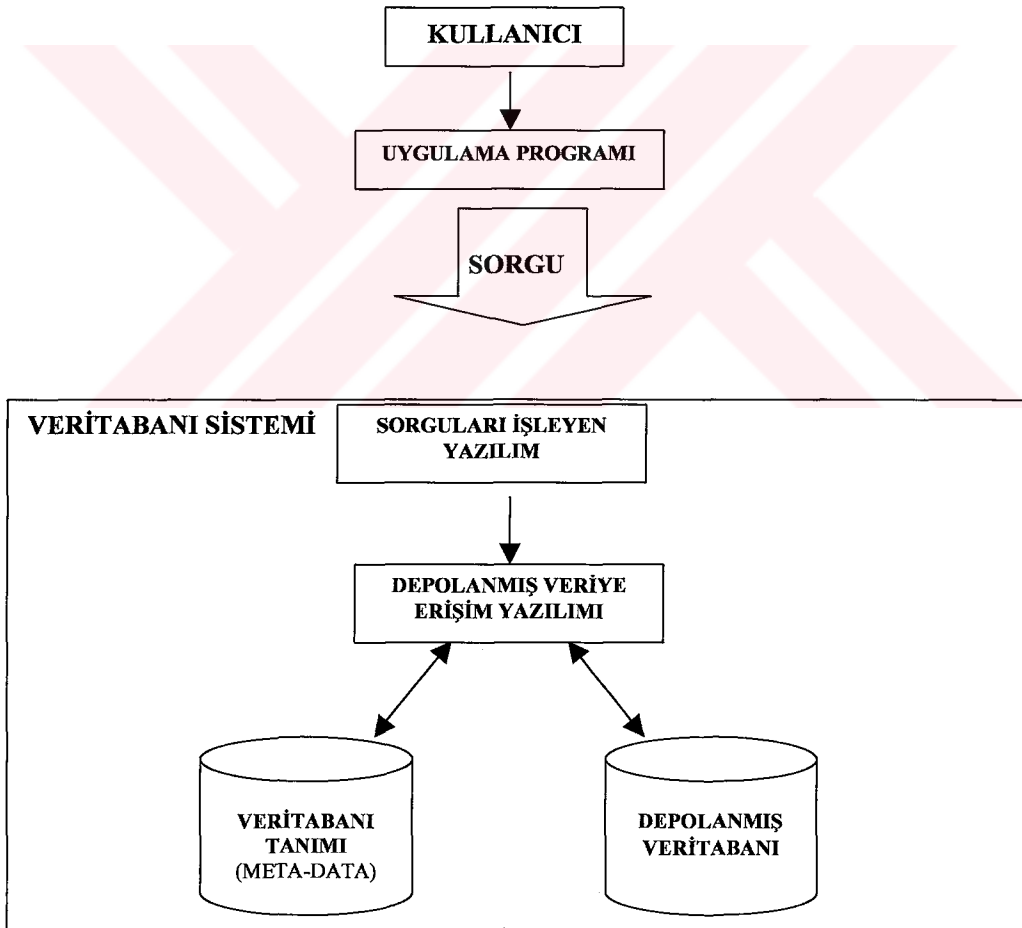
- Şema ve fiziksel organizasyon modifikasyonu: Tanımlamalarla ve bunların DDL derleyicisi ile derlenmesi sonucu şema ve fiziksel organizasyon modifiye edilir.

- Veri erişiminde yetkileri ayarlamak: DBA veri erişimini ayarlar Veritabanının belli kısımlarının belli kullanıcı grupları tarafından görülmesine izin verir. Bu yetkilendirme bilgisi farklı ve özel bir sistem yapısı ile saklanır.

- Kısıtlarının belirtilmesi: Veritabanındaki veri değerlerinin belli tutarlılık kısıtlarına uyması gerekir. Örneğin bir işçinin 1 haftada çalışabileceği saat sayısı için (örneğin 80 saat) gibi bir limit tanımlanır. Her veri girişinde ve güncellemede bu kontrol edilir. [10]

3.1.4 Veritabanı Sisteminin Çalışması

Veritabanı sistemlerinin kullanımı sırasında gerçekleşen aşamalar şekil 3.1 ile ifade edilebilir.



Şekil 3.1 Veritabanı kullanımı sırasında gerçekleşen aşamalar [9]

3.1.4.1 Veritabanı Kullanıcıları

Bir veritabanı sistemi ile amaçlanan, veri çıkarımı yapma, veriye erişme veya yeni veri depolamayı sağlamaktır. Veritabanı sistemi kullanıcıları, sistemle etkileşimlerindeki beklentilerine göre 4 gruba ayrılabilir:

Uygulama Programcıları: Cobol, C, Pascal gibi programlama dillerini kullanan bilgisayar uzmanlarıdır. Yazdıkları programlara, uygulama programları denir ve bu programlar aracılığı ile, veritabanı üzerinde veriler ile değişik işlemler yapılır.

Gelişmiş Kullanıcı: Program yazılmadan sistem ile iletişim kurabilir. Bir sorgulama dili kullanarak istekte bulunurlar. Her bir sorgulama, DML (Data Manipulation Language) ifadelerini, depolama yönetiminin anlayacağı talimatlar haline getirildiği bir sorgu süreci sunar.

Özelleştirilmiş Kullanıcı: Bunlar özelleştirilmiş veritabanı uygulamaları yazan bilgi sahibi kullanıcılarıdır. Bu uygulamalar, klasik dosya sistemine uyum sağlayamayacak çalışmalardır. Örnek olarak bilgisayar destekli tasarım sistemleri, uzman sistemler, grafik ya da işitsel veri yapıları saklayan sistemler ve modelleme sistemleri verilebilir.

Genel Kullanıcılar: Sıradan kullanıcılarıdır. Bir banka veritabanına erişerek, hesap aktarım işlemini yapmak isteyen kişi bu tür kullanıcıdır.

3.1.4.2 Uygulama Programları

Veritabanı ile kullanıcı arasında görev yapan arayüz programlarıdır. Kullanıcının veritabanı kayıtlarına rahatça erişebilmesini, veri ekleme, değiştirme, silme gibi işlemleri rahat yapabilmesini sağlayan programlardır. Bu programlar Delphi, Visual Basic, Visual Fox Pro...gibi çeşitli programlama dilleri ile hazırlanmış olabilir.

3.1.4.3 Meta-Data

Bir veritabanı sistemi, veritabanı bilgileri yanında, aynı zamanda veritabanı sisteminin bir tanımını da içerir. Bu tanım, veritabanı sistemi içinde bulunan her dosyanın yapısını, her veri kaleminin depolanma formatı, veriler üzerindeki çeşitli kısıtlar gibi bilgiler içerir ve bu saklanan bilgi meta-data olarak adlandırılır.

Veritabanının yapısını tanımlayan bu bilgiler bir katalog şeklinde sistem katalogunda saklanır.

DBMS yazılımı, belirli bir uygulamaya özgü olan bir yazılım değildir, genel bir pakettir. Bu nedenle belirli bir veritabanındaki dosyaların yapısını bilmek için sistem kataloguna bakmak gerekir. Bu sayede erişilecek verinin tipi ve biçimi belirlenir.

DBMS yazılımı, sistem katalogunda gerekli bilgiler depolandığı takdirde, herhangi bir veritabanı ile uyum içinde çalışabilmelidir.

3.1.4.4 Sorgu Prosesi

Veritabanında bir tablo göz önüne alalım. Bu tabloda bazı nitelik alanları olacaktır. Örneğin bir ‘çalışanlar’ tablosunda, çalışan no, ad, soyad, doğum tarihi, işe alma tarihi, çalışan numarası, ünvan gibi alanlar bulunabilir. Tablo 3.1’de örnek bir ‘çalışanlar’ tablosu yer almaktadır.

Tablo 3.1 Veritabanında oluşturulmuş ‘çalışanlar’ tablosu

Çalışan No	Ad	Soyad	DoğumTarihi	İşe Alma Tarihi	Çalışan Numarası	Unvan
1	Suzan	Aksoy	05.10.1978	12.12.1991	156	Mühendis
2	Kemal	Kaya	11.09.1974	06.09.1983	056	Mühendis
3	Altan	Galip	03.07.1972	19.11.1993	036	Teknik El.
4	Melike	Türktan	21.12.1976	28.03.1995	112	Sekreter
5	Doğan	Kızıl	10.04.1973	30.11.1999	143	Mühendis
6	Fehmi	Güngör	05.02.1971	24.05.1982	018	Mühendis

Bu tablodan sorgu yolu ile bir bilgi çıkarımı yapmak için bir SQL ifadesi yazılır.

Genel bir SQL ifadesine şu şekilde örnek verilebilir:

SELECT {Alan2, Alan3, Alan7} FROM {TABLO}

Bu durumda, yukarıdaki tablo için,

“ SELECT ad, soyad, ünvan FROM Çalışanlar ”

ifadesi ile çalışanlar tablosundan belirtilen alanların gösterildiği bir liste çıkarılabilir.

Tablo 3.2’de bu liste görülmektedir.

Tablo 3.2 Sorgu sonucunu gösteren liste

Ad	Soyad	Unvan
Suzan	Aksoy	Mühendis
Kemal	Kaya	Mühendis
Altan	Galip	Teknik El.
Melike	Türktan	Sekreter
Doğan	Kızıl	Mühendis
Fehmi	Güngör	Mühendis

Bir veritabanı yönetim sisteminin verileri sorgulama işlemi sırasında şu adımlar sırasıyla gerçekleşmektedir:

“1. Kullanıcı, veritabanına erişimi sağlayan sorgulama programı aracılığı ile yukarıdaki gibi bir SQL komutunu yazarak çalıştırır.

2. DBMS bu komutu alır ve analiz eder.

3. Veri sözlüğünden fiziksel veritabanı tasvirlerini ve gerekli kayıtları içeren disk dosyalarının yerlerini öğrenir.

4. İşletim sistemi devreye girerek, gerekli kayıtları disk üzerindeki dosyalardan alır ve onları sistem bufferlarına aktarır.

5. DBMS veriyi bufferlardan okuyarak kullanıcının önüne getirir.

6. Kullanıcıya istediği liste sunulmuş olur.” [9]

3.1.5 İlişkisel Veritabanı Yönetim Sistemleri (Rdbms)

Veritabanı sistemleri için 3 farklı model belirlenmiştir. Veritabanı yönetim sistemlerinin en önemli sınıflandırması veri modeline göre yapılan sınıflandırmadır.

- a) Hiyerarşik
- b) Ağ
- c) İlişkisel tabanlı veritabanı yönetim sistemlerinden bahsedilebilir.

İlişkisel veritabanı modeli esnek bir model olduğundan birçok ilişkisel veritabanı yönetim sistemi (RDBMS) yazılımı geliştirilmiştir. Bu sistemler, veri modeli olarak ilişkisel tabanlı sistemlerdir. Sistemdeki ilişkisellik hem verilerin etkin kullanımını sağlar, hem de ilişkisel bütünlük kuralı sağlanır.

RDBMS özelliđi, istemci / sunucu veritabanı yönetim sistemlerinde bir standart haline gelmiştir. İstemci / Sunucu veritabanı sistemlerinde işlemler mantıksal olarak, uygulamanın kullanıcı arabirimi ve veritabanı yönetimi olarak ikiye bölünür. İstemci bilgisayar kullanıcı arabirimidir, sunucu ise veritabanı yönetimini kontrol eden birimdir.

Sunucu birimi genellikle Server bir işletim sistemi yazılımı üzerinde çalışır. Sunucu birimi veriler üzerindeki işlemleri daha hızlı yerine getirir. Veriler üzerinde, güvenlik ve paylaşımı sağlar. İstemci birim ise verilerin girildiđi, raporların alındıđı birimlerdir.

İstemci / Sunucu veritabanı modelinde veriler üzerindeki arama, sıralama, aritmetik vb gibi temel işlemler sunucu üzerinde yapılır. Bunun aksine istemci / sunucu olmayan diđer veritabanı uygulamalarında ise verilerin tamamı herhangi bir işlem için istemci bilgisayara taşınacağından istemci bilgisayar ile sunucu bilgisayar arasındaki ağ trafiđi çok daha fazla olur. Bu yüzden performans düşer.

Bunlar göz önüne alındıđında, RDBMS'lerin istemci / sunucu veritabanı yönetim sistemlerinde kullanımının avantajları çoktur. Öncelikle veri bütünlüğü sağlanır. Veri bütünlüğü sunucu düzeyinde gerçekleştirilir. Böylece veriler üzerinde merkezi bir kontrol sağlanır. Ayrıca, ilişkisel veritabanı sistemlerinde verilere erişmek için kullanılan SQL dili, birçok RDBMS geliştiricisi tarafından standart bir dil olarak kabul edilmiştir. Bir diđer avantajı esnekliktir. Yani, veritabanının oluşturulması ve deđiştirilmesi işlemlerinin kolaylıđıdır. Aynı zamanda sistem güvenliđi verilere erişim için gerekli yetki kontrollerinin yapılması ile sağlanmaktadır. RDBMS sistemlerinde güvenlik uygulama bazında deđil, sunucu düzeyinde gerçekleştirilir. En önemli avantajlardan biri de veritabanı içinde gereksiz verilere yer vermemek için öngörülen normalizasyon kurallarını içermesidir.

3.1.6 Veritabanı Dilleri

Bir veritabanı sistemi, 2 farklı tipte dili şart koşar. Biri veritabanı şeması belirlemek için diđeri de, veritabanı sorgulama ve güncelleme kullanımları içindir.

3.1.6.1 Veri Tanımlama Dili (DDL)

Veritabanı şeması, DDL ismi verilen özel bir dil ile ifade edilmiş tanımlamalar kümesi şeklindedir. DDL derleyicisi ürünü, veri sözlüğü adı verilen ve özel dosyalar halinde saklanan tablolar kümesidir. Veri sözlüğü meta-datayı içeren dosyadır. Bu dosya veritabanı sisteminde, asıl verilerin okunması veya değiştirilmesinden önce başvuru dosyadır. Veritabanı sistemleri tarafından kullanılan depolama-saklama yapısı ve erişim metotları, özel bir DDL dili ile ifade edilen tanımlamalar kümesi şeklinde net olarak ifade edilir. Bu tanımlamalar derlenerek, veritabanı şemasının detaylarını belirleyen bir komut kümesi çıkar.

3.1.6.2 Veri Yönetim Dili (DML)

Veriyi yönetmek aşağıdakileri işlemleri kapsar:

- Veritabanında saklanmak üzere verinin eklenmesi
- Veritabanında saklanan verinin yenilenmesi
- Veritabanında saklanan bazı kayıtları silmek
- Veritabanında saklanan verinin değiştirilmesi

Fiziksel seviyede, veriye verimli erişime izin veren algoritmalar tanımlamalıdır. Veri kullanımının daha yüksek seviyelerindeki kolaylığı açıktır. Amaç, sistem ile kullanıcı arasında verimli bir etkileşim oluşturmaktır. DML, kullanıcıların uygun veri modeli ile düzenlenmiş veriye erişimini ve veri yönetimini sağlayan bir dildir. Temel olarak iki tipi vardır.

1. Yordamsal DML'ler : Bir kullanıcının ne verisine ihtiyaç olduğunu ve o veriyi nasıl alacağını belirtmesi gerekir.

2. Yordamsal Olmayan DML'ler : Bir kullanıcının veriyi nasıl alacağını belirtmesine gerek duyulmadan ne verisine ihtiyaç olduğunu belirtmesi yeterlidir.

Yordamsal olmayan DML'leri öğrenmek ve kullanmak, yordamsal olanlara göre daha kolaydır. [10]

Sorgulama, bilgi elde etmek, çıkarım yapmak üzere istekte bulunmaktır. DML'nin bilgi eldesi için kullanılan kısmına 'sorgulama dili' denir. Teknik olarak doğru

olmasa da pratikte sorgulama dili ve veri yönetim dili terimleri aynı anlam için kullanılmaktadır.

3.2 VERİTABANI SİSTEMLERİNE SOYUT YAKLAŞIM

3.2.1 Veri Görünümü

DBMS, kendi içinde birbirleriyle ilişkili dosyalar ile, bunları kullanıcıların erişimine sunan ve bazı işlemlere izin veren programların bir bütünüdür. Bir veritabanı sisteminin en önemli amacı, kullanıcılara kayıtlı verinin özet görünümlerini sunabilmesidir. Burada sunulan bilginin özet olması ile kastedilen, kullanıcı tarafından görülmek istenen, gereksiz detaylardan arınmış bir bilgi olmasıdır. Bilginin nasıl depolandığı, ve işlemlerin nasıl sürdürüldüğüne dair detaylar gizlenir.

3.2.2 Veri Soyutlama

Sistemin kullanışlı olması için, etkin veri kullanımı sağlanmalıdır. Bu da insanları, veritabanındaki veriler ve aralarındaki ilişkileri daha kompleks veri yapıları ile ifade edebilecekleri tasarımlara yöneltmiştir. Bir veritabanı görünümünün, kullanıcıların karşılaşacağı 3 seviyesinden bahsedilir.

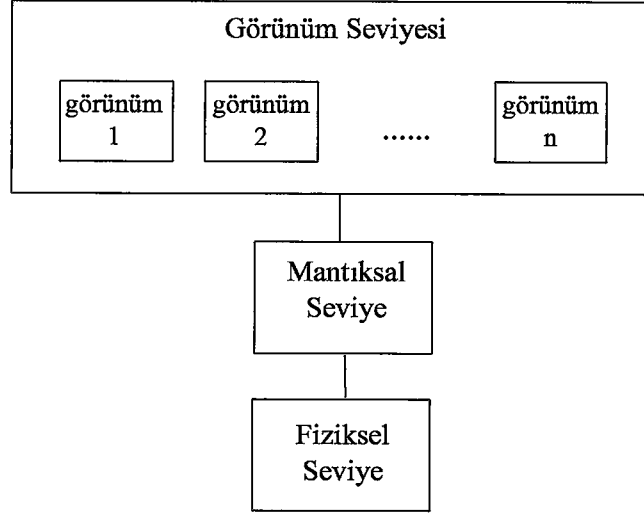
- Fiziksel Seviye
- Mantıksal Seviye
- Görünüm Seviyesi

Fiziksel Seviye: Bilgi gösteriminde en düşük seviyedir ve verinin nasıl saklandığını gösterir. Bu seviyede, veri yapıları detaylı olarak anlatılır.

Mantıksal Seviye: Fiziksel seviyenin bir üst seviyesidir. Bu seviyede veritabanında hangi bilgilerin saklandığı ve bu veriler arasında nasıl ilişkilerin yer aldığını gösterilir. Bu seviye, karmaşıklığın giderildiği, ve sadeleştirilerek detaylara girilmeden veritabanı mantığının anlatıldığı seviyedir.

Görünüm Seviyesi: Bu seviye, veritabanından istenen bazı kısımların gösterilmesidir. Mantıksal seviyenin sadeliğine göre biraz daha karmaşık olabilir. Bu karmaşıklık, veritabanının boyutu ile ilgili olacaktır.

Bu seviyeler arasındaki ilişki şekil 3.2’te gösterilmiştir.



Şekil 3.2 Görünüm seviyesi, mantıksal seviye ve fiziksel seviye arasındaki ilişkiler [10]

Örneğin bir bankanın veritabanında hesap, müşteri, çalışan gibi bazı kayıt tipleri olabilir. Bu kayıt tipleri varlık isimleridir ve bunların çok sayıda örnekleri bulunacaktır. Bu kayıt tiplerinden her birinin nitelikleri için alanlar ayrılacaktır. Örneğin, ‘müşteri’ kayıt tiplerinden birine örnektir. Bu kayıt tipi için belirlenen alanlar, müşteri no’su, ismi, adresi, doğum tarihi gibi alanlar olabilir. Bunların her biri her kayıt için uygun tip veriler ile doldurulacaktır.

Fiziksel seviyede, müşteri, hesap, çalışan kayıtları, kelimeler veya baytlar ile anlatılır. Bu derecede detay programcılardan bile saklanır.

Mantıksal seviyede her kayıt türü ve bunlar arasındaki ilişkiler anlatılır. Bir programlama dili kullanan programcılar bu seviyede çalışırlar.

Bilgisayar kullanıcıları ise, bir dizi uygulama programını görürler ve bu da görünüm seviyesidir. Bir veritabanından pek çok görünüm çıkabilir. Bu seviye kullanıcıların sadece belli veritabanı bilgilerine ulaşmasına izin veren mekanizmayı da sağlar. Örneğin müşteriler, çalışanların maaş bilgilerini görmeyecekleri bir görünüm bulacaklardır.

3.2.3 Veri Modelleri

Bir veritabanı yönetim sistemini, kavramsal ve mantıksal düzeyde temsil etmek üzere veri modeli kullanılır. Veri modellerinde, veritabanı nesneleri, bu nesnelerin nitelikleri, nesneler arası ilişkiler, kısıtlamalar gibi mantıksal kavramlar kullanılır. Veri modeli kullanmanın yararı, kullanıcılar ve uygulama geliştiriciler için, bilgisayarda bilgi depolama ile ilgili teknik ayrıntıları gizleyerek, sadece mantıksal yapıyı sunmasıdır. Veri modelleri aşağıdaki şekilde sınıflandırılabilir:

1. Nesne Tabanlı Mantıksal Modeller
2. Kayıt Tabanlı Mantıksal Modeller
3. Fiziksel Modeller

1. Nesne Tabanlı Mantıksal Modeller:

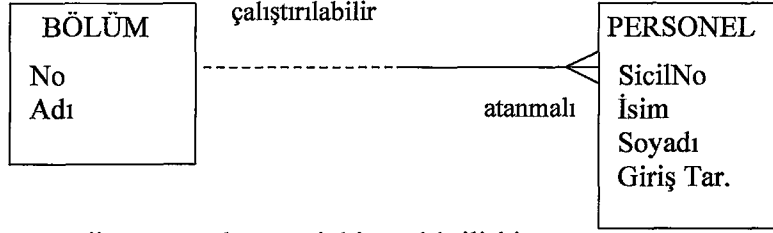
Nesne tabanlı mantıksal modeller, mantıksal ve görünüm seviyesinde veri anlatmak için kullanılır. Gerçekten esnek yapısal olanaklar sağlamaları ve veri kısıtlarının açık ve net olarak belirtilmesi önemli karakterleridir. Birçok farklı model vardır ve daha da artacak gibi gözükmemektedir. En yaygın olarak tanınanları:

1. Varlık-İlişki Modeli
2. Nesne Yönelimli Model
3. Semantik Veri Modeli
4. Fonksiyonel Veri Modeli

Varlık-İlişki Modeli:

E-R veri modeli gerçek dünyayı algılama şeklimize benzer. Temel nesnelere ‘varlık’ adı verilir ve model, bu nesneler ile aralarındaki ‘ilişki’ lerden oluşur. Veritabanında bulunan bilgilere de bu varlıklara ait ‘nitelik’ ler denilir.

Veritabanının tüm mantıksal yapısı, E-R diyagramları ile gösterilebilir. Şekil 3.3’te iki varlık, nitelik bilgileri ve ilişkileri E-R diyagramında görülmektedir.



Şekil 3.3 E-R diyagramından örnek bir varlık-ilişki yapısı

Nesne Yönelimli Model:

E-R modelinde olduğu gibi nesne yönelimli model de nesneleri temel alır. Nesneler ile nesne üzerinde işlem yapabilen kodlar içerir. Bu kodlara metotlar denir. Nesneler aynı tip değerleri ve aynı metotları, “sınıflar“ adı altında gruplar. Sınıflar nesneler için bir tür tanımlama olarak görülebilir. Verilerden ve metotlardan oluşan bu tip tanımlama bir programlama diline benzer.

Bir nesnenin, diğer bir nesnenin verisine erişebilmesinin tek yolu, diğer nesneye ilişkin metodu istemesidir. Bu işleme, nesneye mesaj yollamak denir. Bir nesnenin metotlarına erişmek ve istekte bulunmak, o nesnenin görülebilen kısmı ile ilgilidir. Nesnenin iç kısmı ise (değişken örnekleri ve metot kodları) dışarıdan görülemez.

2. Kayıt-Tabanlı Mantıksal Modeller:

Kayıt tabanlı mantıksal modeller, verinin mantıksal ve görünüm seviyesinde anlatımıdır. Hem veritabanının tüm mantıksal yapısını açıkça belirler hem de uygulama için daha yüksek seviyeden bir anlatım sağlarlar.

Kayıt tabanlı modeller pek çok şekilde isimlendirilir, çünkü veritabanı pek çok türde formatla yapılandırılmış olabilir. Her kayıt tipinin, alanların veya niteliklerin sayısı, ve her alan için de genel bir uzunluk vardır. Pek çok nesne tabanlı modelin tersine kayıt tabanlı mantıksal modelin yapısal zenginliği, fiziksel seviyede değişen uzunlukta kayıtlara imkan verir. Bu sayede, veritabanı işlemlerini yürütmeye kolaylık sağlanır.

En geniş anlamda, kabul gören 3 kayıt tabanlı veri modeli, ilişkisel model, ağ modeli ve hiyerarşik modellerdir. İlişkisel model son senelerde, diğerlerinden daha üstün tutulmakta ve yaygın olarak kullanılmaktadır.

İlişkisel Model:

Veri ve bu veriler arasındaki ilişkileri temsil eden bir çok varlıktan oluşmaktadır. Her varlık, kendisini çeşitli biçimlerde tanımlayan niteliklere sahiptir ve her varlık kendisini diğer örneklerden ayırt etmek için bir veya daha fazla niteliğin bileşiminden oluşan eşsiz tanımlayıcıya sahiptir. Varlıklar fiziksel veritabanında tablo olarak ifade edilirler. Nitelikler ise sütunlar olarak yer alırlar. Eşsiz tanımlayıcılar, birincil anahtar ve ilişkiler de yabancı anahtara dönüşür. Tablo 3.3'te müşteriler, tablo 3.4'te de hesaplar ile ilgili bilgileri saklayan tablolar görülmektedir.

Tablo 3.3 'Müşteriler' tablosu

İsim	Soyadı	İl	Hesap No
Halit	Bozdağ	Konya	1246898
Uğur	Dönmez	Maraş	4578921
Aziz	Arıkan	İzmir	1258679
Remzi	Öz	Bursa	9831672
Yiğit	Gökdemir	Adıyaman	6245008
Orhan	Çiçek	İstanbul	2568931
Uğur	Dönmez	Maraş	1248901
Remzi	Öz	Bursa	4578921

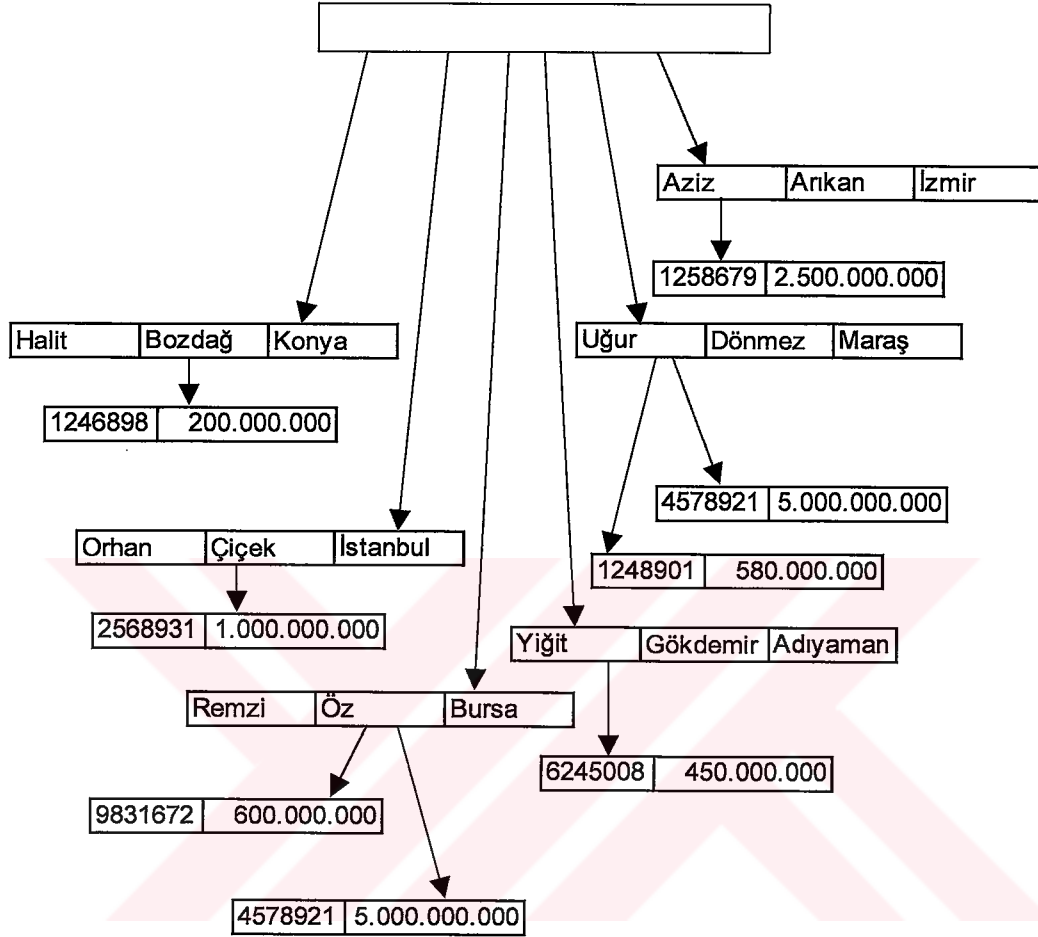
Tablo 3.4 'Hesaplar' tablosu

Hesap No	Bakiye
1246898	200.000.000
4578921	5.000.000.000
1258679	2.500.000.000
9831672	600.000.000
6245008	450.000.000
2568931	1.000.000.000
1248901	580.000.000

Hiyerarşik Model:

Bu model, veri ve ilişkilerin, kayıtlar ve linkler ile gösterilmesi açısından, ağ modeline benzemektedir. Farkı ise kayıtların rasgele çizgeler şeklinde değil, ağaç yapısında organize edilmesidir. Adından da anlaşılacağı gibi bu model hiyerarşik bir

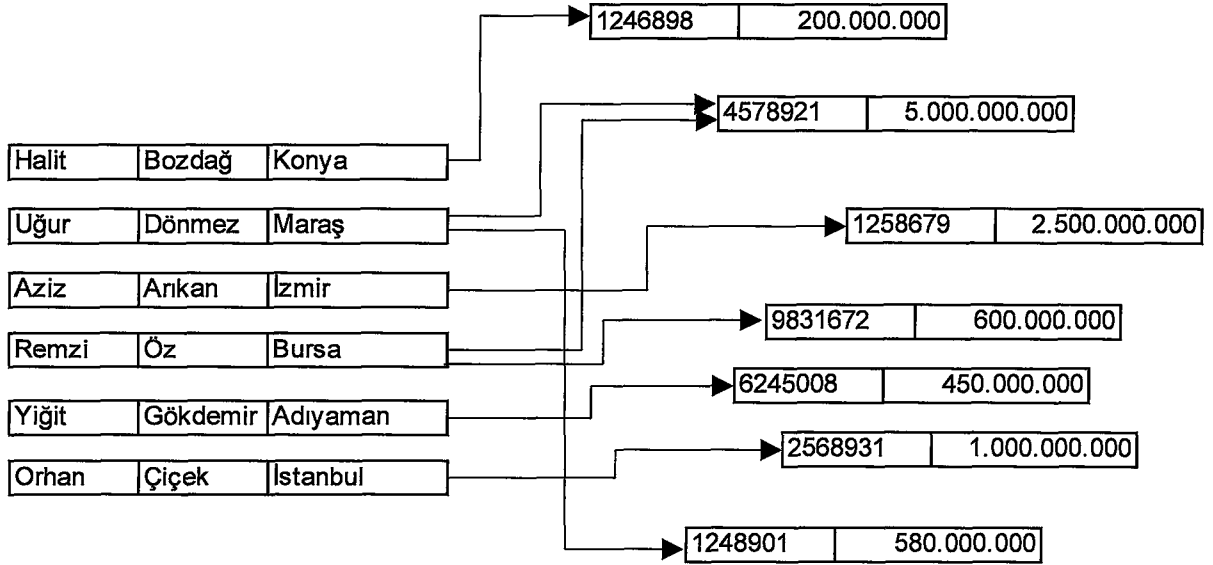
yapıya sahiptir. Sahip-üye, ebeveyn-çocuk ilişkisine dayanan bir modeldir. Yukarıdaki tablolarda verilen bilgiler için hiyerarşik veri modeli olarak şekil 3.4'te gösterilmiştir.



Şekil 3.4 Hiyerarşik veri modeli yapısı

Ağ modeli:

Bu modelde veriler, kayıtlar ile gösterilir ve ilişkiler de linkler ile anlatılır. Bu ilişkiler bir gösterge ile görüntülenir. Veritabanındaki kayıtlar, çizgelerin bir bütünü olarak düzenlenmiştir. Hiyerarşik modelde bazı iyileştirmeler ve değişiklikler yapılarak ortaya çıkan bir modeldir. Burada bir önceki tabloda yer alan bilgiler ağ modeli ile gösterilmiştir. Şekil 3.5'te yukarıdaki veriler ile düzenlenen ağ modeli görülmektedir. [10,12]



Şekil 3.5 Ağ veri modeli yapısı

Modeller Arası Farklılıklar:

İlişkisel model, ağ modeli ve hiyerarşik modelden gösterge veya link kullanılmaması yönüyle ayrılır. Bunun yerine, ilişkisel modelde kayıtlar içerdikleri değerlere göre bağdaştırılır. Bu göstergelerden bağımsız olma özelliği de, ilişkisel veri modeli yapısının resmi bir matematiksel temel ile tanımlanabilmesine olanak tanır.

3. Fiziksel Veri Modelleri:

Fiziksel veri modelleri, en düşük seviyede veri anlatmada kullanılır. Mantıksal veri modellerinin tersine, kullanımda olan az sayıda fiziksel veri modeli vardır. En çok bilinenler unifying model ve frame-memory modelidir.

3.3 SİSTEM VE VERİTABANI GELİŞTİRME

3.3.1 Sistem Geliştirme



Şekil 3.6 Sistem geliştirme aşamaları [9]

Şekil 3.6’da sistem geliştirme aşamaları görülmektedir.

“Strateji safhasında işletme gereksinimlerine uygun bir şekilde; projede izlenecek yol, kullanılacak teknoloji, nasıl bir veritabanı sistemi kullanılacağı (örneğin ilişkisel) gibi temel unsurlar belirlenir.

Analiz aşamasında işletme gereksinimleri tam olarak belirlenir ve bu gereksinimler doğrultusunda kullanılacak veriler ve bu verilerin fonksiyonları ayrıntılı bir şekilde toplanır ve organize edilir.

Tasarım aşamasında ise analiz aşamasında belirlenen ayrıntılı veriler ve bu veriler ile ilgili fonksiyonlar, detaylı bir şekilde veri yapıları şeklinde bilgisayar ortamına aktarılır. Veri yapıları, geliştirilen sistemi en belirleyici şekilde özel programlama dilleri, DBMS gibi bilgisayar platformlarında kullanılacak şekliyle temsil eder.

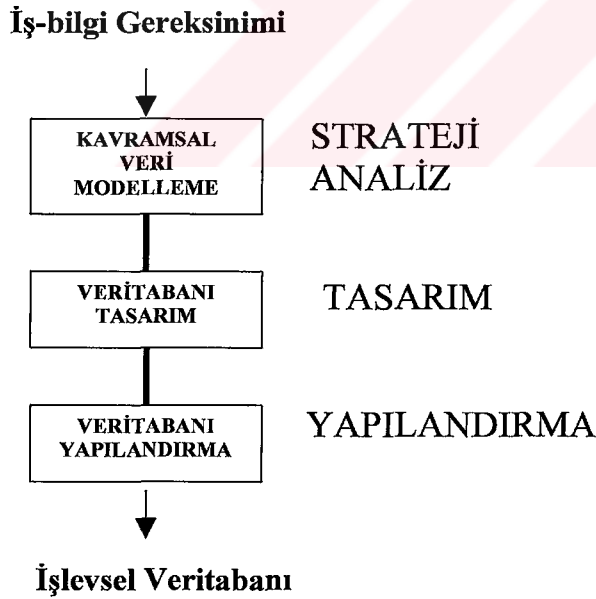
Yapılandırma aşamasında artık program yazılmaya başlanır, formlar düzenlenir, tasarım aşamasında oluşturulan veri yapılarına göre veritabanı ve veritabanı nesneleri oluşturulur.

Dokümantasyon aşamasında; kullanıcı belgeleri, dokümanları, prosedürleri sistemin kullanımını desteklemesi için geliştirilir.

Geçiş aşamasında ise geliştirilen uygulama; kullanıcıların testleri, veritabanı dönüşümü ve buna benzer operasyonlar ile bir ürün halini alır. Son aşamada geliştirilen yazılım işletilir, performansı test edilir ve sistemin bakımı yapılır.

Veri modelleme ve veritabanı tasarımı, sistem geliştirme şemasının ilk üç aşamasına (Strateji, Analiz, Tasarım) dayanır.” [9]

3.3.2 Veritabanı Geliştirme



Şekil 3.7 Veritabanı geliştirme aşamaları [9]

“Kavramsal veri modelleme veya veri modelleme, veritabanı geliştirme sürecinin ilk adımını oluşturur. Ve sistem geliştirme aşamalarından olan strateji ve analiz esnasında gerçekleştirilir.

Veritabanı geliştirme safhalarından biri olan tasarım aşamasında ise oluşturulan veri modeli bir ilişkisel veritabanı tasarımına dönüştürülür.” [9]

3.4 VERİ MODELLEME

3.4.1 Veri Modeli Oluşturma

3.4.1.1 Varlıklar

Varlıkları Belirleme:

Veri modeli oluşturulurken, öncelikle gereksinimleri kavramak, hakkında bilgi tutmaya ihtiyaç duyulan varlıkları belirlemek ve tanımlamak çok önemlidir. İlk aşamada, varlıklar belirlenmeli, tanımlanmalı ve daha sonra model bu temelin üzerine inşa edilmelidir.

Varlık kavramı ile ilgili olarak dikkate alınması gereken bilgiler şunlardır:

Varlık hakkında bilgi depolanması gereken önemli bir nesne ya da kavramdır. Ancak farklı organizasyonlar veya farklı projeler için tanımlanacak varlıklar değişecektir.

Varlık, soyut veya somut olabilir. Yani varlık bir nesne ya da kavramdır. Varlık bir sınıf ya da kategori gibi kendine ait örnekleri içeren bir gruptur.

Örnek olarak, bir işletme ele alındığında, “PERSONEL”, “BÖLÜM”, “MÜŞTERİ”, “SİPARİŞ” varlık olarak tanımlanabilir. Çünkü, büyük ihtimalle işletmenin bunlar hakkında bilgi tutmaya ihtiyacı olacaktır. Bu varlıklardan ilk üçü somut, sonuncusu ise soyut varlıklara örnektir. Ayrıca bu varlıkların her biri bir sınıftır ve kendine ait örnekler içermektedir. Örneğin işletmenin her çalışanı “PERSONEL” varlığının bir örneğidir.

Varlıkları İsimlendirme:

Veri, modelinde kullanılmak üzere belirlenen varlıkların isimlendirilmesi de önemli bir aşamadır. İsim koyarken, bazı noktalara dikkat edilmesi, ileride karşılaşılabilecek sorunları önceden önleyecektir.

Varlıkları isimlendirirken dikkat edilmesi gereken noktalar şunlardır:

Varlığın ismi o veri modeli içinde tek olmalıdır. İki varlığın ismi aynı olamaz.

Varlıkları isimlendirirken karışıklıkları önlemek açısından farklı varlıklara çok benzer isimler vermekten kaçınılmalıdır.

Varlık isimleri anlamlı olmalıdır, ancak mümkünse kısa isimler kullanmak yararlı olacaktır.

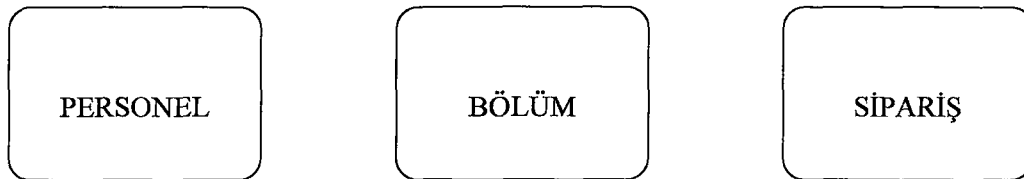
Kullanıcıların bildiği, tanıdık olduğu sözcükler seçilmelidir.

Varlık isimleri cins isim olmalıdır.

Varlık Diyagramları:

Varlıkları tanımladıktan ve isimlendirdikten sonra, o varlığı, varlık-ilişki diyagramında göstermek üzere bir şekil ile ifade etmek gerekir. Varlık-ilişki diyagramlarında, her varlık için bir varlık çerçevesi çizilir. Bu çerçeveleri birbirine bağlayan ilişkiler ile çerçevelerin içlerine de varlığı ilgilendiren nitelikler eklenerek diyagram tanımlanacaktır. Çizilen varlık çerçevelerinin içlerine büyük harflerle varlık isimleri yazılır.

Örnek olarak, bir işletme için belirlenen “PERSONEL”, “BÖLÜM” ve “SİPARİŞ” varlıkları ve onları temsil eden varlık diyagramları şekil 3.8’de göstermiştir.



Şekil 3.8 Varlıkların E-R diyagramında gösterimi

3.4.1.2 Nitelikler

Nitelikleri Belirleme:

Veri modellemede varlıklar belirlenip isimlendirildikten sonra, bu varlık kapsamına girecek örnekler hakkında edinilmek ve saklanmak istenen bilgilerin neler olduğuna karar verilmelidir. Bu bilgilerin hepsi nitelik olarak adlandırılan ve bir varlık örneğinin kimliğini saptamaya, sınıflandırmaya, hakkındaki bilgileri düzenlemeye ve durumunu belirlemeye yarayan öğelerdir. Örneğin “BÖLÜM” varlığı için belirlenen ‘numara’, ‘ad’, ‘yer’ nitelikleri, her bir bölüm örneği için tanımlayıcı bilgiler sağlayacaktır.

Nitelikleri İsimlendirme:

Nitelik isimleri seçilirken dikkat edilmesi gereken noktalar vardır.

Nitelik isimleri, varlığı niteleyecek bilgilerin ismi olmalıdır.

Aynı nitelik ismi, bir varlık için iki kez kullanılmamalıdır. Mümkünse, bütün bir veri tabanı modelinin içinde bile tek olacak şekilde seçilmelidir.

Nitelik isimleri anlamlı ve mümkün olduğunca kısa olmalıdır. Ait olduğu varlığın ismini içermesine gerek yoksa içermemelidir. Örneğin “BÖLÜM” varlığının nitelikleri olarak ‘bölüm numarası’, ‘bölüm adı’, gibi isimler seçmek yerine nitelikleri ‘numarası’, ‘adı’ gibi isimlendirmek daha uygun olacaktır.

Niteliklerin Diyagramda Gösterimi:

Bir varlığa ait nitelikler belirlendikten sonra, varlık-ilişki diyagramında gösterilebilir. Nitelik isimleri hangi varlığa aitlerse o varlık kutucuğunun içine küçük harfler ile yazılırlar. Nitelik isimlerinden bazıları gerekli ve bilinmesi zorunlu bilgilerdir. Bu nitelikler eksik kalmaz. Bu tür nitelik isimlerinin başlarına (*) işareti konarak gerekli oldukları belirtilir. (o) işareti ise özelliğin seçime bağlı olduğunu gösterir.

Şekil 3.9’da, “PERSONEL” varlığı için ‘numarası’, ‘adı’ ve ‘soyadı’ bilinmesi zorunlu nitelikler, ‘boy’, ‘kilo’ nitelikleri ise seçime bağlı nitelikler olarak gösterilmişlerdir.



Şekil 3.9 Bir varlığa ait niteliklerin diyagramda gösterimi

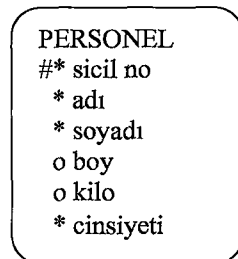
3.4.1.3 Eşsiz Tanımlayıcı Atama

Eşsiz tanımlayıcılar, bir varlığın her bir örneği için farklı olan, örnekleri birbirinden ayırabilecek niteliklerdir. Bir varlık için, bir nitelik eşsiz tanımlayıcı olabileceği gibi, eşsiz tanımlayıcı olarak birden fazla nitelik de belirlenebilir. Örnek olarak “ÖĞRENCİ” varlığı için, kullanılabilecek eşsiz tanımlayıcı “Öğrenci numarası” olabilir, veya “PARÇA” gibi bir varlık için ‘seri numarası’ gibi bir eşsiz tanımlayıcı seçilmesi çok uygun olacaktır.

Eşsiz Tanımlayıcıların Diyagramda Gösterimi:

Eşsiz tanımlayıcı olan bir nitelik, varlık kutucuğunun içinde o özelliğin başına (#) simgesi konması ile belirtilir. Varlık çerçevesinde, maddeler halinde yazılan niteliklerden, eşsiz tanımlayıcı olanı en üste yazılır. Eşsiz tanımlayıcı bir nitelik, aynı zamanda bilinmesi zorunlu bir nitelik olduğundan her zaman (*) simgesine sahip olacaktır.

Şekil 3.10’da, bir “PERSONEL” varlığı nitelikleri, bu niteliklerin gerekleri ve eşsiz tanımlayıcısı gösterilmiştir.



Şekil 3.10 Bir varlığa ait eşsiz tanımlayıcının E-R diyagramında gösterimi

Birden fazla eşsiz tanımlayıcı bulunmasını gerektirecek bir örnek olarak “ÖĞRENCİ” varlığı düşünülebilir. Şekil 3.11’de iki eşsiz tanımlayıcısı olan bir varlık görülmektedir.

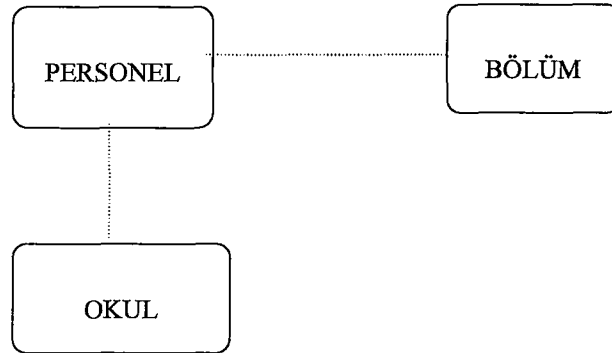


Şekil 3.11 Bir varlık için iki eşsiz tanımlayıcının gösterimi

3.4.1.4 İlişkiler

İlişkileri Anlama:

Sadece varlıkları gösteren bir diyagram fazla açıklayıcı değildir. Çerçevelerin aralarında bir bağlantı veya çerçeve içeriği hakkında herhangi bir açıklama olmadığında diyagram pek bir şey ifade etmez. Ancak her varlık arasında da bir ilişki olması gerekmez. İlişki tanımlarken öncelikle ilişkinin ifade edilmeye değer bir anlam ifade edip etmediğine karar verilmelidir. Bazı varlıklar arasında dolaylı ilişkiler vardır ve bu durumda ilişki tanımlamak gereksiz olacaktır. Örnek olarak “PERSONEL”, “OKUL”, “BÖLÜM” varlıkları göz önüne alınırsa, bu üç varlık çerçevesi arasına ilişki çizgileri çizmeye gerek yoktur. Çünkü “OKUL” ve “BÖLÜM” arasındaki ilişki dolaylı bir ilişkidir. Bu durumda bu üç varlık ve aralarındaki ilişkiler şekil 3.12’deki gibi bir diyagramla temsil edilebilirler.



Şekil 3.12 Varlıklar arası ilişkilerin gösterimi

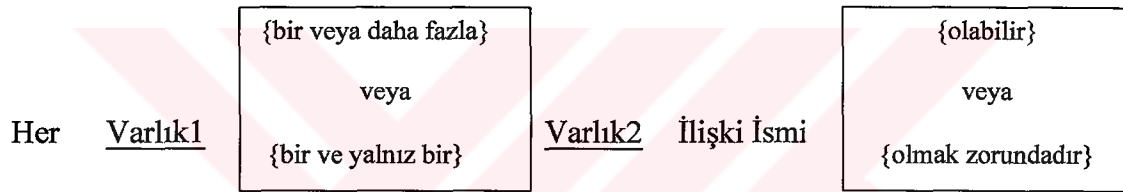
İlişki Cümleciklerinden Yararlanma:

İlişkilerin anlaşılması aşamasında ilişki cümlecikleri kurulması çok yararlıdır. İlişkilerin, ilişki cümlecikleri yardımı ile formüle edilmesi sonucu daha sağlıklı varlık-ilişki diyagramları çizilebilir.

Örneğin “PERSONEL” ve “BÖLÜM” varlıkları arasındaki ilişki şu cümleler ile ifade edilebilir:

- Her bir PERSONEL bir ve yalnız bir BÖLÜM ‘e atanmalıdır.
- Her bir BÖLÜM bir veya daha fazla PERSONEL çalıştırabilir.

Bu tip cümleler incelendiğinde yapının hep aynı olduğu görülür. Cümlelerdeki sözdizimi şekil 3.13’teki formülasyon ile ifade edilebilir:

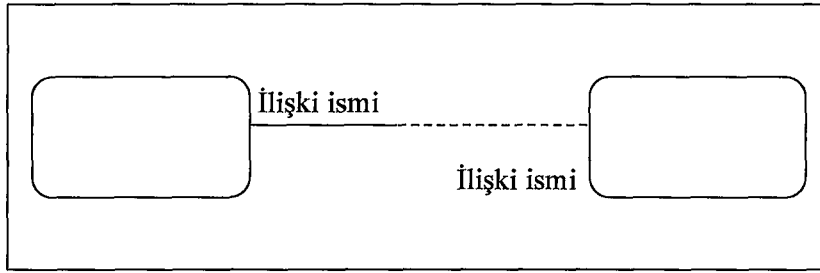


Şekil 3.13 İlişki cümleciklerinin yapısı

Yukarıda görülen yapıdaki iki cümle ile ifade edilen durumda, “PERSONEL” ve “BÖLÜM” varlık belirten kelimelerdir. İlişki isimleri ‘atanmak’ ve ‘çalıştırmak’ tır. ‘olabilir’ ve ‘olmak zorundadır’ sözcükleri ilişkinin zorunlu veya seçime bağlı olup olmadığını, ‘bir veya daha fazla’ ve ‘bir ve yalnız bir’ sözcük grupları da ilişkinin derecesini gösteren kelimelerdir.

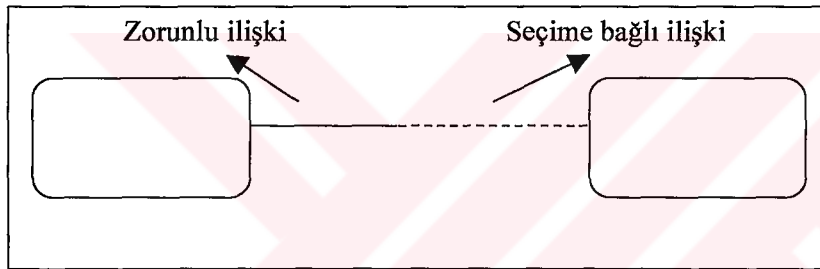
İlişkileri Diyagramla Gösterme:

İlişki kurulmasına karar verilip, kurulan ilişki, ilişki cümlecikleri kullanılarak da netleştirildikten sonra, varlık-ilişki diyagramlarında temsil edilebilir. Bir diyagramda ilişki isimleri, iki varlık arasındaki çizginin uçlarında yer alır. İlişki isimleri yazılırken genelde küçük harfle kullanılır ve saat yönünde okunur. Şekil 3.14’te ilişkilerin genel gösterimi yer almaktadır.



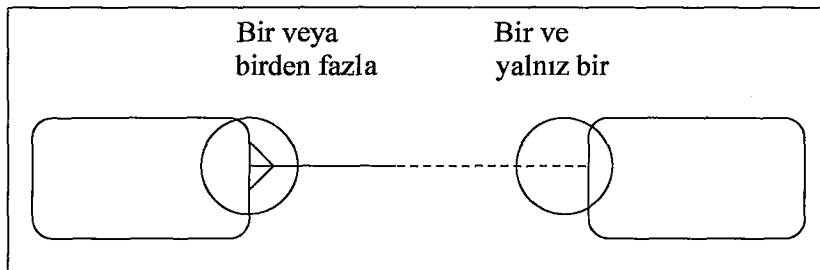
Şekil 3.14 İlişkilerin genel gösterimi

Diyagramda ilişkinin gerekliliği de belirtilir. Zorunlu bir ilişkiyi göstermek üzere sürekli ve düz bir çizgi kullanılır. Seçime bağlı bir ilişkiyi göstermek için ise kesikli çizgi kullanılır. Şekil 3.15'te zorunlu ilişki ve seçime bağlı ilişkilerin nasıl gösterildiği görülmektedir.



Şekil 3.15 Zorunlu ve seçime bağlı ilişkilerin gösterimi

Ayrıca diyagramda ilişkinin derecesi de temsil edilir. İlişki derecesi, çizginin bitiş şekline anlaşılar. İlişki çizgisinin ucu üç tane çizgiden oluşuyorsa, bu ilişkinin derecesinin “bir veya birden fazla” olduğunu gösterir. Eğer ilişki çizgisinin ucunda üç çizgi yok ise bu ilişki derecesi “bir ve yalnız bir” dir. Şekil 3.16'da ilişki derecelerinin gösterimi yer almaktadır.



Şekil 3.16 İlişki derecelerinin gösterimi

İlişki Tipleri:

İlişki tipleri zorunlu ilişki ve seçimli ilişki olarak ayrılır.

Zorunlu ilişkiler kesin ifadelerle anlatılan ve diyagramda düz çizgi ile gösterilen ilişkilerdir. Zorunlu ilişki ifadesi olan ‘olmak zorundadır’ cümlesi ile tanımlanan bir ilişki de o varlığa ait her örneğin o ilişkiye uyması gerekir. Seçimli ilişki ise ‘olabilir’ kelimesi ile ifade edilir, ve örneklerden bazıları ifade edilen ilişkiye sahip olabilir veya olmayabilir.

İlişki Derecesi:

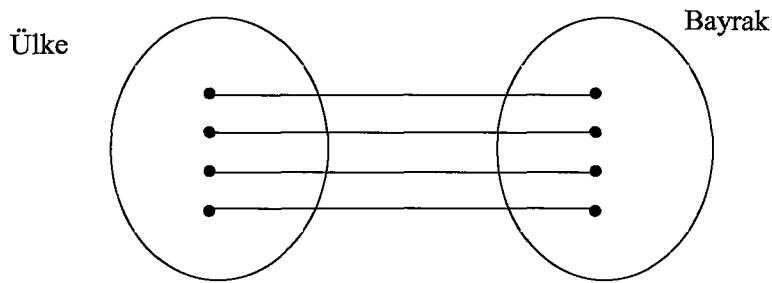
Varlıklar arasında üç çeşit ilişki tanımlanabilir.

- birden-bire ilişkiler
- birden-çoğa veya çoktan-bire
- çoktan-çoğa

Bu ilişkileri ve nitelikleri incelendiğinde şu şekilde özetlenebilir:

Birden-Bire İlişkiler:

“ÜLKE” ve “BAYRAK” varlıkları arasındaki ilişki birden-bire ilişkilere örnek gösterilebilir. Şekil 3.17’de bu tip bir ilişkinin yapısı görülmektedir.



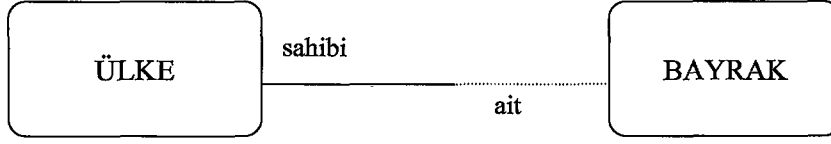
Şekil 3.17 Birden-bire bir ilişkinin yapısı

Burada bir varlığın örneklerinden her biri, diğer varlığın tek bir örneği ile eşleşmektedir.

“ÜLKE” ve “BAYRAK” varlıkları arasındaki ilişkiyi ifade eden ilişki cümlecikleri şu şekilde kurulabilir:

- Her bir ÜLKE bir ve yalnız bir BAYRAK a sahip olmak zorundadır.
- Her bir BAYRAK bir ve yalnız bir ÜLKE ye ait olmak zorundadır.

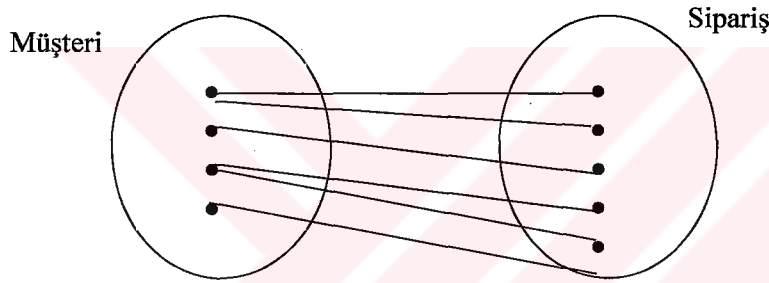
Varlık-ilişki diyagramı ise şekil 3.18’de gösterilmiştir.



Şekil 3.18 Birden-bire bir ilişkinin varlık-ilişki diyagramında gösterimi

- Birden-Çoğa veya Çoktan-Bire İlişkiler:

Bu tip ilişkilere örnek olarak, “MÜŞTERİ” ve “SİPARİŞ” varlıkları arasındaki ilişki şekil 3.19’deki gibi gösterilebilir.

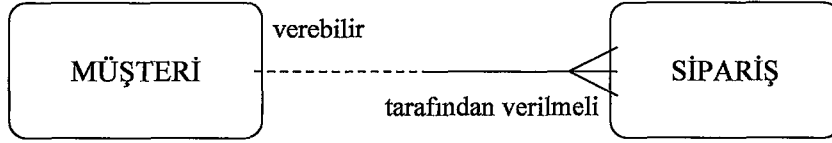


Şekil 3.19 Birden-çoğa veya çoktan-bire ilişkilerin yapısı

Bu ilişkide müşterilerden birinin bir veya birden çok sipariş sahibi olabileceği açıktır. Ancak siparişlerin her biri, bir ve yalnız bir müşteriye sahip olabilir. Bu durumu formüle eden ilişki cümlecikleri şunlardır:

- Her bir MÜŞTERİ, bir veya daha fazla SİPARİŞ verebilir.
- Her bir SİPARİŞ, bir ve yalnız bir MÜŞTERİ tarafından verilmelidir.

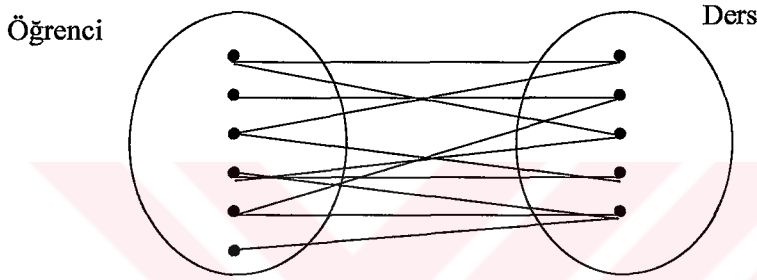
Bu şekilde ifade edilen ilişkinin E-R diyagramında şekil 3.20’deki gibi gösterilecektir.



Şekil 3.20 Birden-çoğa veya çoktan-bire ilişkilerin varlık-ilişki diyagramında gösterimi

- Çoktan-Çoğa İlişkiler:

“ÖĞRENCİ” ve “DERS” varlıkları arasındaki ilişki çoktan çoğa bir ilişkidir. Şekil 3.21’de bu ilişkinin yapısı görülmektedir.

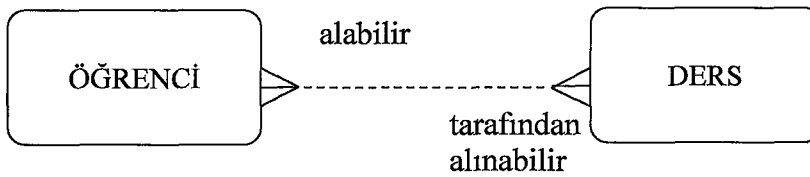


Şekil 3.21 Çoktan-çoğa ilişkilerin yapısı

Her “ÖĞRENCİ” bir veya birden fazla “DERS” alacaktır ve her “DERS”e de kayıtlı bir veya birden fazla “ÖĞRENCİ” bulunacaktır. Bu durumu şu şekilde ifade edebiliriz:

- Her bir ÖĞRENCİ bir veya daha fazla DERS alabilir.
- Her bir DERS bir veya daha fazla ÖĞRENCİ tarafından alınabilir.

Varlık-ilişki diyagramında şekil 3.22’deki gibi temsil edilebilir:



Şekil 3.22 Çoktan-çoğa ilişkilerin varlık-ilişki diyagramında gösterimi

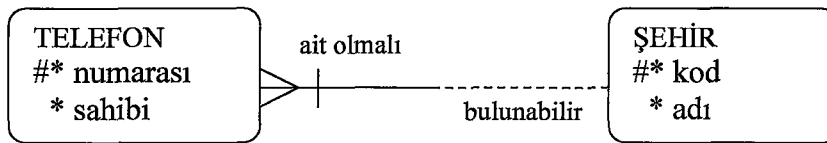
İlişki Yolu ile Gelen Eşsiz Tanımlayıcı:

Eşsiz tanımlayıcı belirlemek konusunda varlıklar arasındaki ilişkiler de önemli unsurlardır.

Örnek olarak bir şehirde herkese 7 haneli telefon numarası verilmekte olsun. Ancak ülke genelinde bu telefon numarasını anlamlı kılabilmek ve tüm numaraları birbirinden ayırmak için, şehir kodları kullanılması gerekmektedir. “TELEFON” varlığı düşünüldüğünde, yalnızca ‘numara’ gibi bir nitelik eşsiz tanımlayıcı olarak seçilirse, bazı durumlarda yeterli bilgi elde edilemeyecektir. Bir şehir içerisinde telefon numaraları tek olmasına rağmen, farklı şehirlerde numaraları aynı olan telefonlar bulunabilir. Bu nedenle “TELEFON” varlığının eşsiz tanımlayıcıları birden fazla olmalıdır ve bunlar da ‘numara’ ve ‘şehir kodu’ olabilir. Bu durumda bir numara kendisi ve şehir kodu ile birlikte anlam kazanacaktır. Şehir kodu “TELEFON” varlığının eşsiz tanımlayıcılarından biri haline gelmiştir ve bu ilişki “tanımlayıcı ilişki” adını alır.

Tanımlayıcı İlişkilerin Diyagramla Gösterimi:

Yukarıda anlatılan örnekte “TELEFON” varlığının eşsiz tanımlayıcıları o telefonun numarası ve şehir kodudur. Burada ‘numara’ özelliği “TELEFON” varlığının kendi içindedir, ancak diğer eşsiz tanımlayıcı olan ‘şehir kodu’ “ŞEHİR” isimli başka bir varlıktan gelmektedir. İlişki çizgisinin önüne kısa ve dik bir çizgi çekilerek, bu tür ilişkiler gösterilebilir. Şekil 3.23’te, örneğe ait varlık-ilişki diyagramı yer almaktadır.



Şekil 3.23 Tanımlayıcı bir ilişkinin varlık-ilişki diyagramında gösterimi

3.4.2 Veri Modelini Arıtma

3.4.2.1 Veri Modelini Normalize Etme

Normalizasyon, doğru varlığa doğru niteliklerin atanmasını sağlamak amacıyla adım adım gerçekleştirilen işlemler dizisidir. Literatürde 5'e kadar kural tanımlanan fakat genelde 3 ilkedен oluşan normalizasyon gerçekleştirilir. Normalizasyonun 3 ilkesi de bir niteliği bir varlığa atama konusunda kullanılan ilkelerdir.

“Bir niteliğe bir varlığı atarken o niteliğin oraya ait olup olmadığını anlamak için sürekli olarak üç soru sormak gerekir.

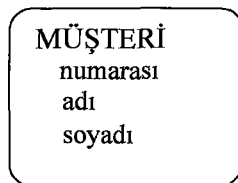
1. Bu nitelik sadece bir kez mi tekrar edecek?
2. Bu nitelik tüm eşsiz tanımlayıcı niteliğe mi bağımlı?
3. Bu nitelik başka bir nitelik üzerine bağımlı değil mi?

Eğer bu 3 soruya da “Evet” cevabı verebiliyorsanız nitelik doğru yerindedir.” [13]

Normalizasyonda kullanılan 3 ilke şu şekilde isimlendirilir:

- Birinci Normal Form
- İkinci Normal Form
- Üçüncü Normal Form

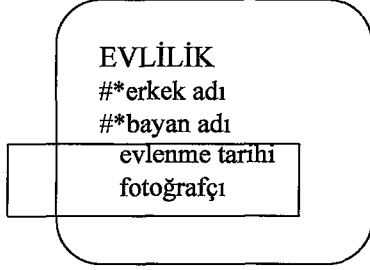
Birinci normal form : Bu ilke, birinci soru ile ilgilidir. Birinci normal formu sağlayabilmek için cevabın “evet” olması, yani bir özelliğin ait olduğu varlığın içinde yalnızca bir kez yer alması gerekir. Şekil 3.24'te bu ilkeye uygun bir varlık görülmektedir:



Şekil 3.24 Birinci normal forma uyan bir varlık diyagramı

Sipariş gibi müşteri için birden fazla kere tekrar edebilecek şeyler nitelik olarak seçilmemelidir. Bu tür bir seçim, birinci normal formun ihlal edilmesi olacaktır.

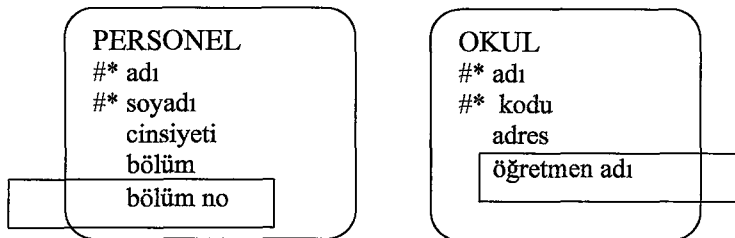
İkinci normal form : Bu ilke, ikinci soruya “evet” cevabı verebilmeyi gerektirir, yani tanımlanan nitelikler eşsiz tanımlayıcıların tümüne bağlı olmalıdır. Örneğin “EVLİLİK” varlığı göz önüne alınsın. Şekil 3.25’te ikinci normal forma uygun bir tasarım görülmektedir.



Şekil 3.25 İkinci normal forma uyan bir varlık diyagramı

Yukarıda “EVLİLİK” varlığının eşsiz tanımlayıcısının tek bir tane olmadığını görülmüyor. Buradaki iki eşsiz tanımlayıcı, ‘erkeğin adı’ ve ‘kadının adı’ dır. Bunların ikisine birden bakılarak bir evlilik ayırt edilebilir. Bu varlık için tanımlanan ‘evlenme tarihi’ ve ‘fotoğrafçı’ niteliklerinin ikisi de bu iki eşsiz tanımlayıcıya da bağlıdır. ‘Evlilik tarihi’ ve ‘fotoğrafçı’, “EVLİLİK” varlığının eşsiz tanımlayıcıları olan ‘erkeğin adı’ ve ‘kadının adı’ niteliklerinin her biri için farklı değildir. Ancak eğer “EVLİLİK” varlığı altına doğum tarihi, boy, kilo gibi eşsiz tanımlayıcılardan teki için tanımlanabilecek nitelikler konulsa idi ikinci ilke sağlanmamış olacaktı ve hatalı bir tasarım yapılmış olacaktı.

Üçüncü normal form : Bu ilke ise üçüncü soruya “evet” cevabını verebilmeyi gerektirir. Aranılan şart, seçilen niteliklerin ait olduğu varlığı doğrudan ilgilendiren bilgiler olmasıdır. Bir varlığın nitelikleri, başka bir varlığın eşsiz tanımlayıcılarına bağlı olmamak üzere dikkatlice belirlenmelidir. Varlık için yanlış belirlenmiş nitelikler sorun çıkaracaktır. Üçüncü normal forma aykırı bir örnek şekil 3.26’da verilebilir:



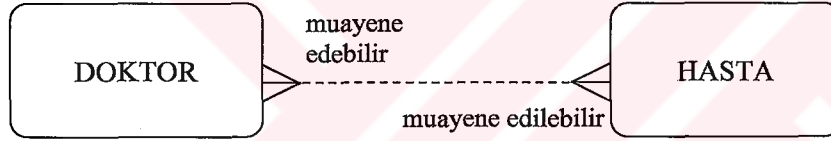
Şekil 3.26 Üçüncü normal forma aykırı varlıklar

“PERSONEL” varlığındaki ‘bölüm no’ özelliği “PERSONEL” varlığının altında bulunmak yerine bölümü ilgilendiren bir nitelik olarak “BÖLÜM” gibi bir varlık altında bulunması uygundur.

“OKUL” varlığındaki ‘öğretmen adı’ özelliği de ‘okul adı’ ve ‘okul kodu’ eşsiz tanımlayıcılarına bağlı bir nitelik değildir. ‘öğretmen adı’ nitelik olarak “ÖĞRETMEN” varlığını ilgilendiren bir nitelik olabilir, ve “ÖĞRETMEN” varlığında tanımlanmış ‘öğretmenlik no’ gibi bir eşsiz tanımlayıcıya bağlı bir nitelik olabilir.

3.4.2.2 Çoktan-Çoğa İlişkileri Çözümleme

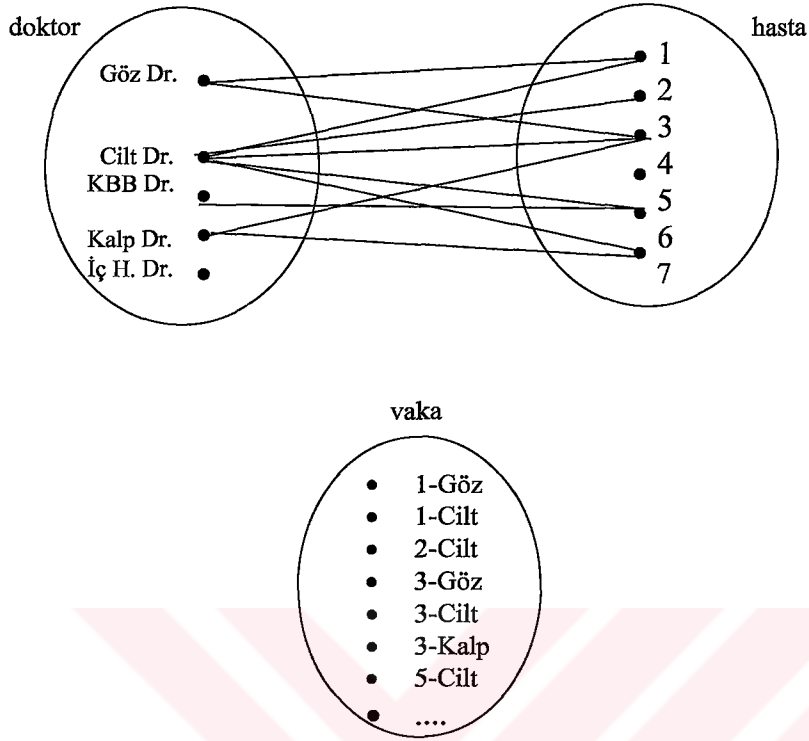
Çoktan-çoğa ilişkiler, her iki yönde de birden fazla değer alan ilişkilerdir. “DOKTOR” ve “HASTA” varlıkları arasındaki ilişki düşünüldüğünde, bir hasta aynı anda birden fazla doktor tarafından tedavi edilebileceği gibi, bir doktor da bir veya birden fazla hastaya bakmaktadır. Bu ilişki şekil 3.27’deki varlık-ilişki diyagramında temsil edilebilir.



Şekil 3.27 Çoktan-çoğa bir ilişkinin varlık-ilişki diyagramında gösterimi

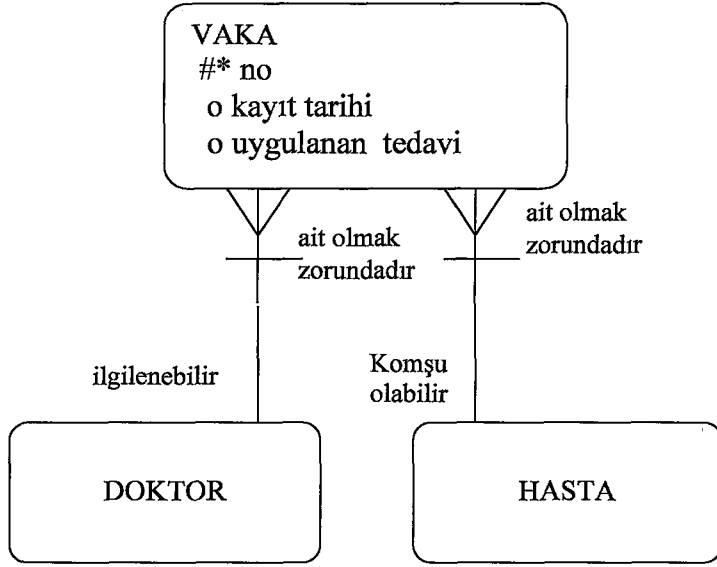
Burada doktor ve hastayı ilgilendiren, hastalığa ve tedaviye dair bilgiler vardır. Bunlar her iki varlığı da ilgilendireceği için ortak niteliklerdir, çoktan-çoğa olan bu ilişkiyi daha basit ve anlamlı kılmak üzere kullanılırlar. Bu tip ilişkileri çözmek için ortak nitelikleri içinde bulunduran bir kesişim varlığı eklenecektir. “DOKTOR” ve “HASTA” varlıklarını birleştiren “VAKA” gibi bir varlık eklemek uygun olacaktır.

Bu durum, şekil 3.28’de gösterildiği gibi ifade edilebilir.



Şekil 3.28 Çoktan çoğa bir ilişkiye eklenen kesişim varlığı

Sonuç olarak, “VAKA” varlığını koyarak veri modellemede bir rahatlama sağlanır. “VAKA” varlığı ise “DOKTOR” ve “HASTA” varlıklarının her ikisini de ilgilendiren niteliklere sahip olacaktır. Örnek olarak ‘kayıt tarihi’, ‘uygulanan tedavi’, ‘sonuç’ gibi nitelikler tanımlanabilir. Bu çözümleme sonucu varlık-ilişki diyagramı şekil 3.29’daki gibi olacaktır:



Şekil 3.29 Bir kesişim varlığı ile çözümlenmiş çoktan çoğa bir ilişkinin varlık-ilişki diyagramında gösterimi

Yukarı diyagramda “VAKA” varlığı ile bu varlığın birleştirdiği varlıklar arasındaki ilişki şöyle okunur:

- Her bir DOKTOR bir veya daha fazla VAKA ile ilgilenebilir.
- Her bir HASTA bir veya daha fazla VAKA konusu olabilir.

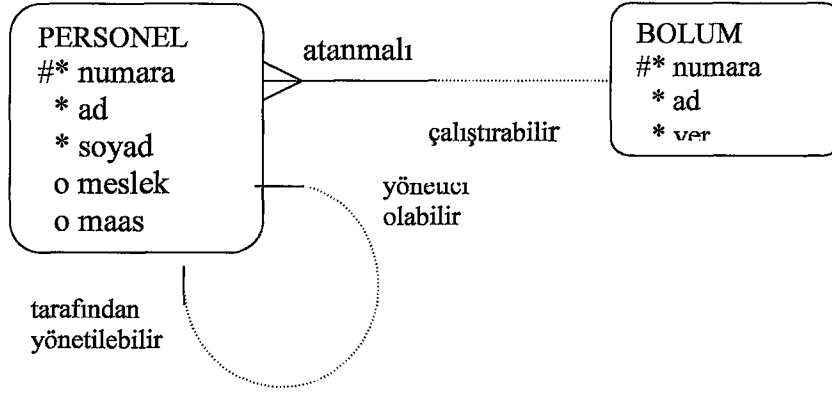
İlişkiler ise “VAKA” varlığından yola çıkılarak şöyle okunabilir:

- Her bir VAKA bir ve yalnız bir DOKTOR a ait olmak zorundadır.
- Her bir VAKA bir ve yalnız bir HASTA ya ait olmak zorundadır.

3.4.2.3 Tekrarlı İlişkileri Modelleme

Tekrarlı ilişki, herhangi bir varlığa ait örneklerden birinin, diğer örnekler ile arasındaki ilişkiyi temsil eder.

Örnek olarak “PERSONEL” varlığı düşünüldüğünde, personel örneklerinden bazılarının yöneticileri yine PERSONEL varlığı içinde bir kişi olabilir, bu durumda diyagrama bir tekrarlı ilişki göstergesi eklenir. Şekil 3.30’da bu tekrarlı ilişkiye ait varlık-ilişki diyagramı görülmektedir.

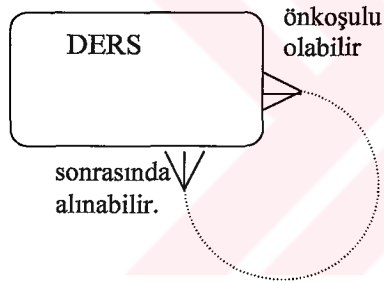


Şekil 3.30 Tekrarlı bir ilişkiye ait varlık-ilişki diyagramı

“PERSONEL” varlığındaki bu tekrarlı ilişki şöyle okunabilir:

- Her bir PERSONEL bir ve yalnız bir PERSONEL tarafından yönetilebilir.
- Her bir PERSONEL bir veya daha fazla PERSONEL ’e yönetici olabilir.

Tekrarlı ilişkinin aynı zamanda çoktan-çoğa bir ilişki olması şekil 3.31’de görülüyor:



Şekil 3.31 Tekrarlı ve çoktan çoğa bir ilişkinin varlık-ilişki diyagramında gösterimi

Bu diyagram şu ilişki cümlecikleri ile ifade edilebilir:

- Her bir DERS bir veya daha fazla DERS in ön koşulu olabilir.
- Her bir DERS bir veya daha fazla DERS sonrasında alınabilir.

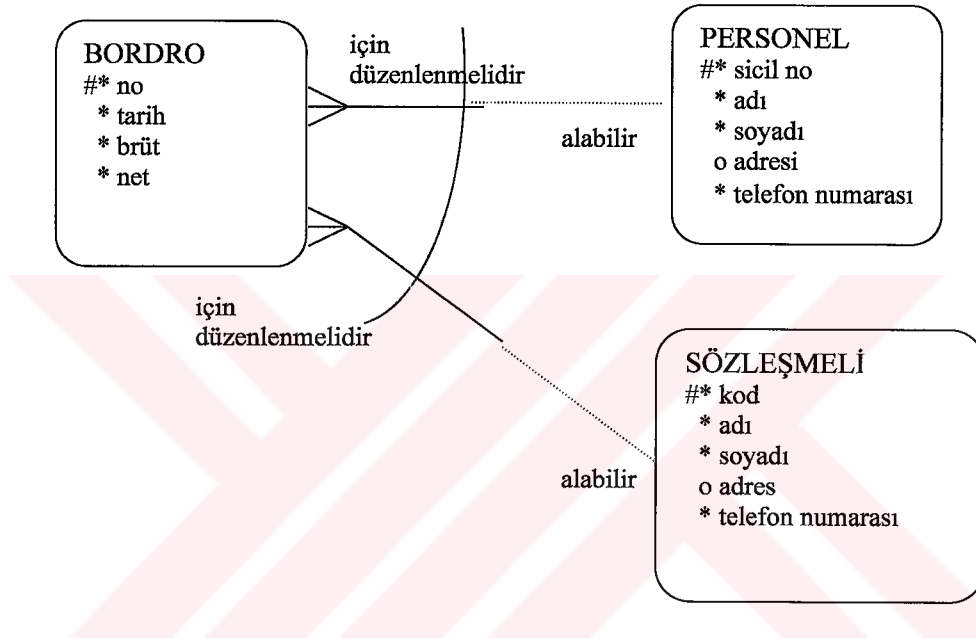
3.4.3 İleri Seviyeli İlişkiler

3.4.3.1 Özel İlişkileri Modelleme

“Özel ilişkiler iki eşdeğer varlık ile bu varlıkların ikisinin birden ilişkili olduğu üçüncü bir varlık arasında bulunur.” [9]

Bir işyerinde, hem personel hem de sözleşmeli çalıştığı düşünölsün. Her iki grup içinde bordro düzenlenmekte olsun. İşyeri hem personel hem de sözleşmeliler için bilgi tutacaktır. “BORDRO” varlığı, PERSONEL ve SÖZLEŞMELİ ile özel bir ilişkiye sahiptir. “BORDRO” varlığının bazı örnekleri “PERSONEL” bazı örnekleri de “SÖZLEŞMELİ” için olacaktır. “BORDRO” varlığının tüm örnekleri ya “PERSONEL” ya da “SÖZLEŞMELİ” için olacaktır.

Bu özel ilişkiyi göstermek üzere bir yay kullanılacak, ve şekil 3.32’deki gibi bir diyagram ortaya çıkacaktır.



Şekil 3.32 Özel ilişkilerin varlık-ilişki diyagramında gösterimi

Çizilen bu yay bordronun, “PERSONEL” veya “SÖZLEŞMELİ” varlıklarından sadece birine ait olabileceğini gösterir. Yani iki ilişki aynı anda aktif olamaz.

Yukarıdaki diyagram, aşağıdaki gibi okunur:

- Her bir BORDRO ya bir ve yalnız bir PERSONEL için ya da bir ve yalnız bir SÖZLEŞMELİ için düzenlenmelidir.
- Her PERSONEL bir veya daha fazla BORDRO nun sahibi olabilir.
- Her SÖZLEŞMELİ bir veya daha fazla BORDRO nun sahibi olabilir.

3.4.3.2 Alt Varlık ve Üst Varlıkları Tanımlama

Bazı veri modellerinde, ortak nitelikleri fazla olan varlıklar ile karşılaşılabilir. Bu tür durumlarda, ortak nitelikleri fazla olan bu varlıkları genel bir varlığın altında alt varlıklar olarak düşünmek mantıklı bir yaklaşım olacaktır.

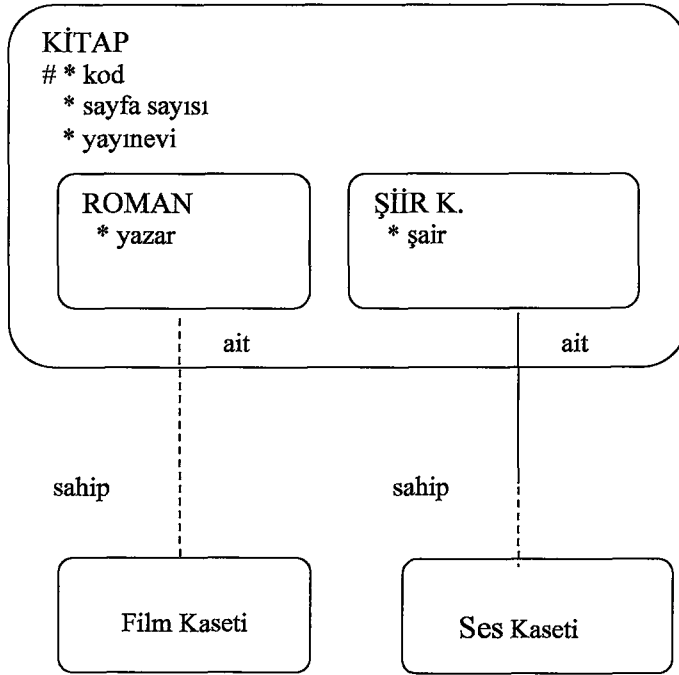
Örneğin, bir mağaza sattığı ürünleri sınıflandırırken “ROMAN”, ve “ŞİİR KİTABI” varlıklarını kitap bölümünün alt grupları olarak sınıflandırabilir. Bu varlıkların satılan diğer ürünlere göre çok ortak nitelikleri bulunmaktadır. Aşağıda gösterildiği gibi, bu varlıklar “KİTAP” varlığı altında toplanabilir. “ROMAN”, ve “ŞİİR KİTABI” varlıklarının ortak nitelikleri, “KİTAP” varlığı kutusunda tanımlanmalıdır.

Diyagram şekil 3.33’te gösterildiği gibi olacaktır:



Şekil 3.33 Alt tip ve süper tiplerin varlık-ilişki diyagramında gösterimi

Üst varlık ve alt varlıkların her ikisinin de başka varlıklar ile ilişkileri olabilir. Şekil 3.34’te bu tip bir ilişkinin diyagramı görülmektedir.



Şekil 3.34 Üst ve alt varlıkların ayrı ayrı başka varlıklar ile ilişkileri olması durumu

3.4.4 Veri Modeli Dönüşümü

Veri modeli, normalizasyon ilkelerine uyularak oluşturulduktan sonra, fiziksel veritabanı tasarımına dönüştürülür.

3.4.4.1 İlişkisel Veritabanı

Veriler veritabanlarında tablolar şeklinde saklanmaktadır. Tablolar bir veya daha fazla sütun ve sayısız satır içerebilirler. Bir tablonun içinde en az bir tane sütun bulunmak zorundadır. Tablodaki satırlar o varlığın örneklerine aittir, ve her sütunda bu örnekler ile ilgili nitelik bilgileri tutulmaktadır. Bir anlamda, sütunlar bu tutulan bilgilerin sınıflarıdır. Tablo 3.5'te örnek bir veritabanı tablosu görülmektedir.

Tablo 3.5 Örnek bir veritabanı tablosu

↓ SÜTUN

ÖĞRENCİ

YOKLAMA NUMARASI	SOYADI	ADI	DOĞUM TARİHİ	ÖĞRENCİ NUMARASI
304	AVCI	MURAT	10.08.1976	1194304
389	UÇA	PINAR	04.11.1978	1196389
542	YAVUZ	HANDAN	01.04.1961	1179542

→ SATIR

“DBMS deki bir tablodan veri alacak herhangi bir program, bilgi alacağı tablonun sütun isimlerini kullanmalıdır ve bu, ilişkisel DBMS’lerindeki en önemli prensiplerdendir.” [9]

Birincil Anahtarlar:

Bir birincil anahtar, bir veya birden daha fazla sütundan oluşur ve tablodaki her kayıt için birbirinden farklı değer alır. Bu şekilde, tablodaki her satır birincil anahtar aracılığı ile birbirinden ayrılır. Her tablo, birincil anahtar bulundurmamak zorundadır. Tablo 3.6’da bir veritabanı tablosu ve üzerinde birincil anahtar olarak kullanılacak sütun gösterilmiştir.

Tablo 3.6 Tablo ve birincil anahtar olarak kullanılacak sütun

**BİRİNCİL
ANAHTAR**

↓

ÖĞRENCİ

ÖĞRENCİ NUMARASI	SOYADI	ADI	DOĞUM TARİHİ	YOKLAMA NUMARASI
1194304	AVCI	MURAT	10.08.1976	304
1196389	UÇA	PINAR	04.11.1978	389
1179542	YAVUZ	HANDAN	01.04.1961	542

Yukarıdaki “SINIF” tablosunda “öğrenci numarası” sütunu tablonun birincil anahtar alanıdır. Bu tablodaki her kayıt, yani her öğrenci örneği, birincil anahtar olan “öğrenci numarası” için farklı değer alır.

Yabancı Anahtarlar:

Bir yabancı anahtar, bir veya birden daha fazla sütundan oluşur ve aynı veya herhangi başka bir tablodaki birincil anahtar alanını referans olarak alır. Tablo 3.7’de fakülte tablosunda kayıtlı olan fakülte kodu kendi tablosu içinde birincil anahtardır ve tablo 3.8’de öğrenci tablosuna yabancı anahtar olarak gelmektedir.

Tablo 3.7 ‘Fakülte’ tablosunda birincil anahtar

↓ BİRİNCİL
ANAHTAR **FAKÜLTE**

FAKÜLTE KODU	ADI
1	ELEKTRİK ELEKTRONİK FAK.
2	MADEN FAK.
3	MAKİNA FAK.

Tablo 3.8 ‘Sınıf’ tablosunda yabancı anahtar kullanılması

ÖĞRENCİ YABANCI ANAHTAR ↓

ÖĞRENCİ NUMARASI	SOYADI	ADI	DOĞUM TARİHİ	FAKÜLTE KODU
1194304	AVCI	MURAT	10.08.1976	3
1196389	UÇA	PINAR	02.10.1949	2
1179542	YAVUZ	HANDAN	01.04.1961	1

“ÖĞRENCİ” tablosundaki “FAKÜLTE KODU” sütunu, görüldüğü gibi “FAKÜLTE” isimli bir başka tablonun birincil anahtar alanına bağlıdır.

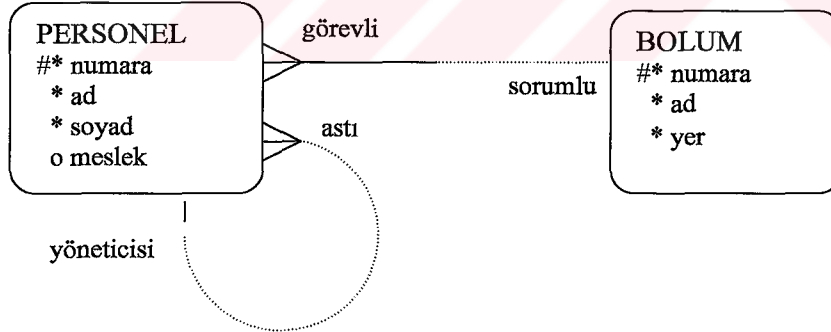
3.4.4.2 Varlık-İlişki Diyagramını Fiziksel Veritabanına Dönüştürme

Kavramsal veritabanı tasarımında kullanılan nesneler bir ilişkisel veritabanı tasarımına aşağıdaki şekilde işlemler yapılarak dönüştürülür. Tablo 3.9’da ilişkisel veritabanı dönüşümünde, varlık-ilişki modelindeki kavram ve nesnelere denk gelen yapılar listelenmiştir.

Tablo 3.9 Varlık-ilişki modelindeki kavram ve nesnelerin ilişkisel veritabanı dönüşümü

Varlık-İlişki Modeli	İlişkisel Veritabanı
Varlık	Tablo
Nitelik	Sütun
Eşsiz Tanımlayıcı	Birincil Anahtar
İlişki	Yabancı Anahtar
Zorunlu nitelik	Not null kısıtı

Örnek olarak şekil 3.35’teki E-R modeli verilmiş olsun.



Şekil 3.35 Örnek bir E-R diyagramı

Yukarıdaki E-R diyagramındaki varlıklar ve nitelikler bir ilişkisel veritabanı sistemine dönüştürüldüğünde tablo 3.10 ve tablo 3.11’deki gibi temsil edilirler.

Tablo 3.10 ‘Bölüm’ tablosu

BİRİNCİL ANAHTAR			BÖLÜM	
bölüm_no	bölüm_adı	yer		
birincil anahtar				
eşsiz t./zorunlu	zorunlu	seçimli		
1	pazarlama	Nişantaşı		
2	kalite-kontrol	Kadıköy		

Tablo 3.11 ‘Personel’ tablosu

PERSONEL

personel_no	adı	soyadı	mesleği	bölüm_no	yönetici_no
birincil anahtar				yabancı anahtar	yabancı anahtar
eşsiz t./zorunlu	zorunlu	zorunlu		zorunlu	
1	Elif	Uça	işletme müdürü	1	101
2	Ilknur	Kızılay	mühendis	2	
3	Berkant	Yavuz	mühendis	1	
4	Melisa	Türk	insan kaynakları şefi	2	201

Yukarıdaki tablolarda görüldüğü gibi “PERSONEL” tablosundaki “bölüm_no” ve “yönetici_no” sütunları birer yabancı anahtardır. Dolayısıyla “bölüm_no” sütununa girilecek değerler “BÖLÜM” tablosundan ve “yönetici_no” sütununa girilecek değerler de tekrarlı bir ilişki vasıtası ile yine “PERSONEL” tablosundan gelecektir.

“PERSONEL” tablosunun birincil anahtarının “personel_no”, “BÖLÜM” tablosunun birincil anahtarının ise “bölüm_no” isimli sütunlar olduğu görülmektedir.

3.5 SQL (STRUCTURED QUERY LANGUAGE)

3.5.1 SQL Dili

İlişkisel veritabanı modelinin, IBM araştırma laboratuvarlarında çalışmalarını sürdüren Dr. Edgar F. Codd tarafından geliştirilmesi ile SQL ve diğer veritabanı dillerinin gelişmesinde de büyük ilerleme görüldü. 1970 li yıllarda IBM ve Dr. Edgar

F. Codd tarafından geliştirilen SQL, ilişkisel veritabanı sistemlerinde bir standart haline gelmiştir. O zamandan bugüne kadar çeşitli SQL versiyonları çıkmıştır. [14]

Orjinal SQL dili, Kasım 1976 da “IBM Journal of R&D” dergisinde tanıtılmıştır. 1979 yılında ise Oracle Corporation, ticari ilk SQL ortamını tanıtmıştır. [9]

1986 yılında ANSI (American National Standards Institute) SQL dili için bir standart tanımlamış ve bu standart, ISO (International Standards Org.) tarafından 1987 yılında kabul edilmiş ve SQL ortaya çıkmıştır. [11] İlk standart olarak tanımlanan SQL1, yayınlandıktan kısa bir zaman içinde yetersiz kalmış ve 1989 yılında SQL 89 standart olarak yayınlanmıştır. Ardından 1992 yılında SQL2 standardı yayınlanmıştır. Bu standart ANSI SQL ve SQL-92 isimleri ile de anılmıştır. [12]

3.5.2 SQL Komutları

Bir ilişkisel veritabanı modeli için Dr. Codd tarafından tanımlanan 12 kuralda şu temel özellikler verilmiştir:

İlişkisel bir Veritabanı Yönetim Sistemi veritabanlarını tamamen ilişkisel yetenekleri ile yürütebilmelidir. Veritabanı kapsamında içerilen tüm bilgiler tablolardaki değerler olarak ifade edilmelidir. Her bir değere tablo adı, birincil anahtar değeri, ve sütun adı karışımları kullanılarak ulaşılabilmelidir. Sistematik null değer desteği, aktif ilişkisel katalog bulunmalıdır. Anlaşılır ve belli söz dizimine sahip bir veri alt dil desteği olmalıdır. Görünüm güncelleme, küme seviyesinde veri işlemleri özellikleri olmalıdır. Fiziksel erişim yapısı veya kayıt yapıları değiştiğinde sistem etkilenmemeli ve uygulama programları da tablo yapılarındaki değişimden mümkün olduğunca etkilenmemelidir. Veritabanı dili tümleşiklik kurallarını tanımlamaya izin vermeli, katalog içinde erişilebilir tutmalı, ve uygulamalar bunu göz ardı etmemeli, sistem dağıtıklık bağımsızlığına sahip olmalı ve veri dilinden başka dillerde tümleşiklik aşılmamalıdır. [15]

SQL Dili üç kısımda incelenir:

1. Veri Tanımlama Dili
2. Veri İşleme Dili
3. Veri Sorgulama Dili

1. Veri Tanımlama Dili Komutları:

Bu komutlar veritabanını meydana getirecek tablo oluşturma, güncelleme ve silme işlemlerini sağlar. Öncelikle veri modeline göre, tablo oluşturulur. Bu tablolarda öncelikle sütunlara uygun isimler verilmeli ve bu sütunlara girilecek veri tipi ile veri uzunluğuna, ve veri girişinde ‘null’ veya ‘not null’ kısıtına karar verilir. Birincil ve yabancı anahtarlar da belirlendikten sonra veri girişi yapılır. [14,15]

“CREATE TABLE” komutu ile tablo adı, sütun adı ve sütun veri tipi tanımları yapılır.

Tablo isimlendirilirken, veritabanı sisteminin izin verdiği sayıda karakter kullanılır. İlk karakterin alfabetik bir karakter olmasına dikkat edilir. Genellikle İngiliz alfabesi harflerinin kullanılması tavsiye edilmektedir. Belirlenen tablo isimleri, veritabanı sisteminde ayrılmış bir sözcük olamaz.

Bir sütun isimlendirilirken, yine tablo isimlendirmede olduğu gibi veritabanı sisteminin izin verdiği uzunluğu geçmeyecek şekilde, ilk karakter alfabetik bir karakter olmak üzere isimlendirilir.

Sütunlara girilecek veri tipleri belirlenirken ise, şunlar göz önünde bulundurulur [15]:

String Veri Tipi : Karakter ve metin türü veriler için,

Number Veri Tipi : Tamsayı, gerçek sayı gibi hesaplanan tüm veriler için,

Date / Time Veri Tipi : Her türlü tarih ve zaman verileri için,

Blob Veri Tipi : Resim, müzik, film vs. boyutları farklı verileri işlemek için kullanılır.

Örneğin, aşağıdaki ifadeler ile bir tablo oluşturulabilir:

```
CREATE TABLE OKUL  
(OKUL_KOD      INTEGER      NOT NULL,  
OKUL_ADI       VARCHAR (19)  NOT NULL,  
OKUL_ADRESİ    VARCHAR (19)  NOT NULL,  
KR_TARİH       DATE);
```

Veri tanımlama dili komutlarından biri de ALTER TABLE komutudur. Bu komut ile, tablolara sütun ve kısıt eklenebilir, veya veri tipi ve kısıtları yeniden düzenlenebilir.

Örneğin yukarıdaki CREATE TABLE komutu ile oluşturulan tabloya ‘Kurum’ tablosundan bir yabancı anahtar, bir de yeni sütun ilave edilmek istendiğinde aşağıdaki komutlar kullanılacaktır:

```
ALTER TABLE OKUL  
    ADD FOREIGN KEY (KURUM_KOD)  
        REFERENCES (KURUM),  
    ADD MUDUR VARCHAR (19) NOT NULL;
```

Bir diğer veri tanımlama komutu DROP TABLE’ dır. Veritabanında istenmeyen tabloları ortadan kaldırmak için kullanılır. Aşağıdaki komut ile OKUL tablosu veritabanından kaldırılacaktır:

```
DROP TABLE OKUL
```

2. Veri İşleme Dili Komutları:

Veri işleme komutları olarak, INSERT, UPDATE, ve DELETE verilebilir.

INSERT komutu tüm sütunlara değer girecek şekilde, bazı sütunlara değer girecek şekilde, veya bir SELECT sorgusu ile seçilerek değer girilecek şekilde kullanılabilir.

[15]

Aşağıdaki komutlar ile, OKUL tablosuna veri girilebilir:

```
INSERT INTO OKUL (OKUL_KOD, OKUL_ADI, OKUL_ADRESİ)  
VALUES (10032, ‘YUNUSEMRE IOO’, ‘Yunusemre Cad No33’ )
```

UPDATE komutu ile veri güncellenir. Bu komut ile, tek bir satırın veya birden fazla satırın güncellenmesi yapılabilir. Aşağıdaki komut bir satırdaki değerlerden birini güncellemek için kullanılmıştır:

```
UPDATE OKUL
  SET OKUL_ADI= 'YUNUSEMRE ILK'
  WHERE OKUL_KOD= 10032;
```

Yukarıdaki ifade ile, OKUL tablosunda okul kodu 10032 olan okulun okul adı YUNUSEMRE ILK olarak güncellenmek isteniyor.

Diğer veri işleme komutu ise DELETE 'dir. Bu komut, tabloda bir veya birden fazla satırı silmeyi sağlar. Bu komut aşağıdaki gibi uygulanır:

```
DELETE FROM OKUL
WHERE OKUL_KOD=10032;
```

Bu durumda okul kodu verilen okul ile ilgili satır silinecektir.

3. Veri Sorgulama Dili Komutları:

Veri tabanından verileri SELECT komutu ile seçeriz. Bir veya birden fazla sütun seçilebilir, kayıtlarla ilgili belli bilgiler veya tüm bilgi istenebilir. SELECT komutu WHERE kalıpları ile birlikte de kullanılır. Bu durumda, FROM ile belirtilen tablodan WHERE ile belirtilen koşulu sağlayan kayıtlar arasından seçim yapılacaktır. WHERE ile sağlanması istenen koşul bazı karşılaştırma operatörleri ile de belirlenebilir. Aşağıdaki böyle bir sorgulama örneğidir:

```
SELECT OKUL_ADRESİ
  FROM OKUL
  WHERE OKUL_KOD > 10000
```

Bu sorgu ile okul tablosunda, okul kod numarası 10000'den büyük olan kayıtların adreslerine ulaşılabacaktır. Burada WHERE ile kullanılan karşılaştırma operatörleri olarak =, <, >, <=, >=, <> verilebilir. Bunlar sırası ile, eşit, küçük, büyük, küçük veya eşit, büyük veya eşit, eşit değil işaretleri yerlerine kullanılır. Ayrıca BETWEEN, IN, LIKE, CONTAINING, IS NULL, STARTING WITH gibi başka karşılaştırma operatörlerinden de söz edilebilir.

3.5.3 Embedded SQL

SQL'i, C, ADA, PASCAL, COBOL gibi programlama dilleri arasında ve bu dillerle bir arada kullanmak mümkündür. [11,12] Bu da SQL dilinin en önemli kullanımlarından biridir. Bu tür kullanımlarda programın içinde SQL komutları başlamadan önce bir ön ek veya komut yazılarak diğer komutlardan ayrılır. [12]

3.5.4 SQL Server

Dağıtılmış ortamlardaki veri gereksinimi, verilerin sunucu bilgisayarlara taşınmasını zorunlu kılmış ve istemci / sunucu veritabanı ortamlarının doğmasını sağlamıştır. SQL Server veritabanı işlemlerinde yaygın olarak kullanılan bir dil olan SQL dilinin veritabanı sunucu yazılımlarında kullanılması ile SQL Server adını almıştır. SQL Server, verilerin organizasyonunu merkezi olarak yürütürken, istemci uygulamaların sunucu üzerindeki verilere erişmesini sağlar. Bu özellikler uygulamalardan yapılan işlemlerin sunucu üzerinde otomatik olarak devam ettirilmesini ve verilerin bütünlüğünün tesis edilmesini sağlar. [9]

SQL Server istemci / sunucu mimariye sahip bir veritabanı yönetim sistemidir, ilişkisel bir yapıya ve veri işleme araçlarına sahiptir, uygulama geliştirmeye açık bir ortamdır, ve RDBMS özelliği ile verilerin etkin bir şekilde organize edilmesini sağlar. SQL Server 'ın istemci / sunucu mimarisi, iş yükünü sunucu ve istemci üzerinde çalışmak üzere ikiye ayırır. İstemci işlemi yapmaktan sorumludur ve kullanıcıya bilgi sunar. Genellikle birden çok sayıda olur. Sunucu ise veritabanının yer aldığı bilgisayardır ve bellek, disk işlemleri, ağ genişliği olarak daha geniş kaynaklara sahiptir.

Microsoft SQL Server, tam olarak Microsoft Transact SQL'i desteklemektedir. Microsoft SQL Server veritabanı sisteminde, sistem ve kullanıcı olmak üzere iki tür veritabanı vardır. Sistem veritabanları, sistemin kullandığı ve SQL server 'ın yüklenmesi sırasında yükleme programının otomatik oluşturduğu veritabanlarıdır, ve kullanıcı veritabanları da kullanıcıların, SQL Server yüklendikten sonra ihtiyaçları doğrultusunda oluşturdukları veritabanlarıdır.

SQL Server 'ın kuruluđu ile 4 tane sistem veritabanı oluđu. Bunlar kullanıcı veritabanlarını ve SQL Server 'ın işlemlerini kontrol eden ve sistem katalođunu tutan master veritabanı, yeni oluđturulacak kullanıcı veritabanları için prototip oluđturan model veritabanı, geçici tablolar veya bellek gereksinimleri için saklama alanı sađlayan tempdb, SQL Server Agent servisini destekleyerek işlerin programlaması hakkında bilgiler sađlayan msdb veritabanıdır.

Master veritabanı SQL Server veritabanı sisteminin meta-data sınıfı bulunduđu veritabanıdır. Loginleri, sistem düzenlemelerini, diđer sistem veritabanları ve sonradan oluđturulacak olan kullanıcı veritabanları hakkındaki bilgileri tutar.



4. KENT BİLGİ SİSTEMLERİ (KBS)

4.1 KENT YÖNETİMİNDE KBS İHTİYACI

4.1.1 Kent Yönetiminde İhtiyaçlar

“Günümüzde, nüfusun büyük çoğunluğu kentlerde yaşamaktadır. Kentli insanların, yaşam standardı yüksek bir ortamda yaşayabilmeleri için kent yönetiminde başarı sağlamak gerekmektedir. Ancak bu çok kolay değildir, çünkü kent yönetimi çok kompleks bir yapı sergilemektedir. Kent yönetiminde etkin olabilmek için üstün yöneticilik vasıflarının yanı sıra, kullanılacak yöntemler de önemlidir. Sağlıklı, etkin, hızlı, ucuz, verimli hizmet ve çözümler sunabilmek gerekir.” [16]

Bu amaca ulaşmak için teknolojinin getirdiği yeniliklerden yararlanmak zorunludur. Bilgisayar teknolojisi, bu çalışmalarda büyük kolaylıklar sağlar. Bilimsel ve teknik gelişmelerin hayatlarımıza kattığı kolaylıklar sayesinde, cereyan eden olgu ve olayları, coğrafyada bulundukları konumları itibariyle bilmek mümkündür. Aynı şekilde değerlendirme yapmak, ve ortaya çıkacak her türlü probleme kısa sürede ve doğru çözüm üretmek de kolaylaşmıştır. Bilginin mevcut yapıda olduğu gibi manual yöntemlerle üretilmesi, modellenmesi, işlenmesi ve kullanılması zaman ve işgücü kaybına neden olmaktadır.

4.1.2 Bilgi Sistemleri ve Coğrafi Veri

Gelişmiş ülkeler doğru ve güncel bilgiye, kısa sürede ulaşmak için bilgi teknolojilerinden yararlanmaktadırlar. Bilgilerin bir disiplin altında toplanması ve kullanıcıların hizmetine sunulmasının en uygun yolu, bilgisayarlardan istifade ederek bilgi sistemlerinin oluşturulmasıdır. Aslında Bilgi Sistemleri terimi bir uygulamayı, kullanılan veri dosyalarını ve bu sistemi çalıştırmak için kullanılan bilgi teknolojisini içermektedir.

Kentler, günümüzde yoğun hizmet talep eder hale gelmiştir. Pek çok farklı konuda ihtiyaç duyulan bu hizmetler incelendiğinde, bunların büyük bir kısmının coğrafya ile ilgili olduğu görülür. Sosyal koşulların, yaşam seviyelerinin ve yaşama

koşullarının planlanmasında, sağlık, ekonomi, tarım reformu v.b. politikalar hazırlanmasında, izlenmesinde mekanla ilişkili veriler kullanılacaktır. Görülmektedir ki, kent yönetimlerinde kullanılması gereken bilgi sistemleri, coğrafi konum bilgisi içeren verilerle çalışacaktır. Mekansal sorunların çok çeşitli ve karmaşık oluşu, bu konuda çalışan tüm kurum ve kuruluşları çözüme ulaştıracak bir yaklaşıma ve çıkış aramaya yöneltmiştir. Bunun sonucunda da çözüm olarak mekansal bilgi sistemleri oluşturulmaya başlanmıştır. Coğrafi Bilgi Sistemleri de mekansal bilgi sistemlerindendir. Coğrafi Bilgi Sistemleri, coğrafi nesnelere ait grafik ve grafik olmayan coğrafi verilerin kullanıcı ihtiyaçlarını karşılayacak biçimde çeşitli kaynaklardan toplanması, depolanması, işlenmesi, bu verilerin veri tabanı işlemleri, sorgulamalar, dönüşümler ve coğrafi analizler ile coğrafi bilgiye dönüştürülmesi, yönetilmesi sunulması fonksiyonlarını bütünsel olarak yerine getiren yazılım ve personel bileşenlerinden oluşan bir organizasyon, bu veri ve bilgilerin gösterimi için kullanılan gelişmiş bilgi sistemleri olarak tanımlanmıştır. [17]

“CBS’nin diğer bilgi sistemlerinden farkları net olarak görülmektedir. CBS, yeryüzündeki nesneler ve onlarla ilgili öznitelik bilgilerinin kullanıldığı bir sistem olarak Uzaktan Algılama (UA), Bilgisayar Destekli Tasarım (CAD) ve Veri Tabanı Yönetim Sistemlerinden farklıdır. UA Sistemleri uydulardan, hava fotoğraflarından veya scanner adlı özel tarayıcılardan elde edilen raster veriler üzerinde çalışarak yeryüzü ve atmosfer hakkında veriler toplayan, saklayan ve sunan sistemlerdir. Büyük ölçekli çalışmalarda kullanmak için henüz yeterince gelişmemiştir. CAD Sistemleri özel olarak nesnelerin tasarım ve düzenlenmeleri için geliştirilmiş ve veritabanı bağlantıları olmayan veya sınırlı olan sistemlerdir. Bu sistemler CBS göre daha az miktarda veri ile ilgilenirler. DBMS Sistemleri coğrafik olmayan, nesnelere ait öznitelik bilgilerinin saklanması ve istendiği anda hızlı bir şekilde ulaşılması için geliştirilmiş bugün en sık kullanılan sistemlerdir. Bu sistemler bugün halen tek başlarına coğrafik verilerin saklanması ve analizlerinin yapılarak sunulması için yeterli sistemler değildir.” [18]

“CBS Sistemleri aslında bu sistemlerden tamamen bağımsız ve kullandığı hem coğrafi hem de coğrafi nesnelere ait coğrafi olmayan öznitelik bilgileri dolayısıyla bu sistemlerin üzerinde değil de bu sistemlerin hepsinin kesiştiği yerde duran bir bilgi sistemidir. CBS Sistemlerini diğer sistemlerden ayıran temel özellik bu sistem

içerisinde yapılan ve yapılmaya müsait olan ve coğrafi veriler üzerinde yapılabilen analiz kabiliyetidir.” [19]

4.1.3 KBS Kullanımı

Kente ait verilerin coğrafi bileşenleri ile bilgisayar ortamına aktarıldığı, sorgulandığı, denetlendiği ve yönetildiği Kent Bilgi Sistemleri ise CBS'nin özel bir halidir. KBS için pek çok veri toplanır. Farklı amaçlarla, farklı ve ortak pek çok veri kullanılarak son derece karmaşık problemlerin çözümünde kullanılan bu sistem, çeşitli kurumları ilgilendirir. “Kent yönetiminden sorumlu olan pek çok kurum ve kuruluş bulunmaktadır. Bu kurum ve kuruluşlar yatay ve düşey konumda karşılıklı etkileşim ve eşgüdüm içinde, ilişkiler ağına kavuşmalıdır.” [16]

Her kurumun kendi ihtiyaçlarına göre oluşturabileceği veri tabanı sistemlerinin, kurumlar arasında veri alış verişlerinde büyük sorunlar yaratabileceği ortadadır. Birliği sağlayacak standartlar belirlenmelidir. Bilgilerin korunması önemli olduğu kadar bu bilgilere ulaşmak ve yeni bilgiler üretmek de önemlidir. Her kuruluş sorumluluğu altında bulunan verileri üretmeli ve güncel tutarak diğer kurum ve kişilerin hakları ölçüsünde kullanıma sunmalıdır.

Aslında hedef kent boyutunda değil ülke bütününde, tüm kurum ve kuruluşlar için belli standartların belirlenmesi ve ayrıca veri paylaşımı konusunda ortaklık sağlanması olmalıdır. Bu tür bir yaklaşım bir çok yönden bugün yaşanan sıkıntıların giderilmesini sağlayacaktır.

“Genel olarak; dünyanın gelişmiş ülkelerinde insanların, karşılaştıkları sorunları çözebilmek için; yalnızca kurumsal araştırmalar yapmakla kalmayıp, ülkelerin özelliklerini de dikkate alarak, bu ülkelerin imkanlarına uygun gelecek, geniş kapsamlı uygulamalara da yöneldikleri görülmektedir.

Geniş anlamda CBS veya kent bilgi sistemi içinde değerlendirilen bilgiler ve buna dayalı çalışmalar, kurumlara ait verilerin bir merkezi sistem içerisinde toplanmasını gerektirmektedir. Bu da çok sayıda kurumun yakın ilişki ve işbirliğinin yanı sıra, aynı sistem içinde verilerini birleştirmiş olan kurumlar arasında bağlantıyı gerekli hale getirmektedir. Burada sözü edilen bağlantı ile kurumların arasında yapılacak işbirliği ve bilgisayarların birbirine bağlanması ifade edilmektedir.” [17]

Bu tür bir organizasyon ve yapı sağlamak yıllar alacak bir çalışma gerektirecektir. Böyle bir sistemin kurulmasının kısa sürede olması beklenemez. Ancak günümüzde sahip olduğumuz teknoloji ile, en azından şehirlerimizde Kent Bilgi Sistemlerinin kurulması ve bu teknolojiye hiç olmazsa şehir bazında yaşanan problemlerin çözümünde faydalanılması yönünde çalışmalar yapılmalıdır.

4.1.4 KBS Kullanımı ile İlişkili Temel Kuruluşlar

“Kent Bilgi Sistemini kullanacak ve bu veriler ilişkiler ağı içerisinde yer alan temel kuruluşlar :

Valilikler, Belediyeler, Muhtarlıklar

Tapu ve Kadastro Müdürlükleri

Su ve Kanalizasyon İdaresi

Gaz İdaresi

Toplu Taşımacılık İdaresi

Elektrik İdaresi

Telefon İdaresi olacaktır.

Bu kuruluşların yapmış oldukları temel görevler baz alınarak analiz çalışmaları için uygulama programları geliştirilmiştir. Geliştirilen uygulama programları şu konuları içerebilir [16] :

Emlak İstimlak Uygulamaları

Gelirler Uygulamaları

Halkla İlişkiler Uygulamaları

Kültür Uygulamaları

Özel Kalem Uygulamaları

Park Ve Bahçeler Uygulamaları

Teftiş Heyeti Uygulamaları

Temizlik İşleri Uygulamaları

Zabıta Uygulamaları

Hesap İşleri Uygulamaları

Hukuk İşleri Uygulamaları

İdari Ve Mali İşler Uygulamaları

Fen İşleri Uygulamaları

Harita Uygulamaları

İktisat Uygulamaları

İmar Ve Planlama Uygulamaları

Makine Elektrik Ruhsat Uygulamaları

Makine İkmal Bakım Onarım Uygulamaları

Yazı İşleri Ve Kararlar Uygulamaları

Basın Danışmanlığı

Personel Uygulamaları

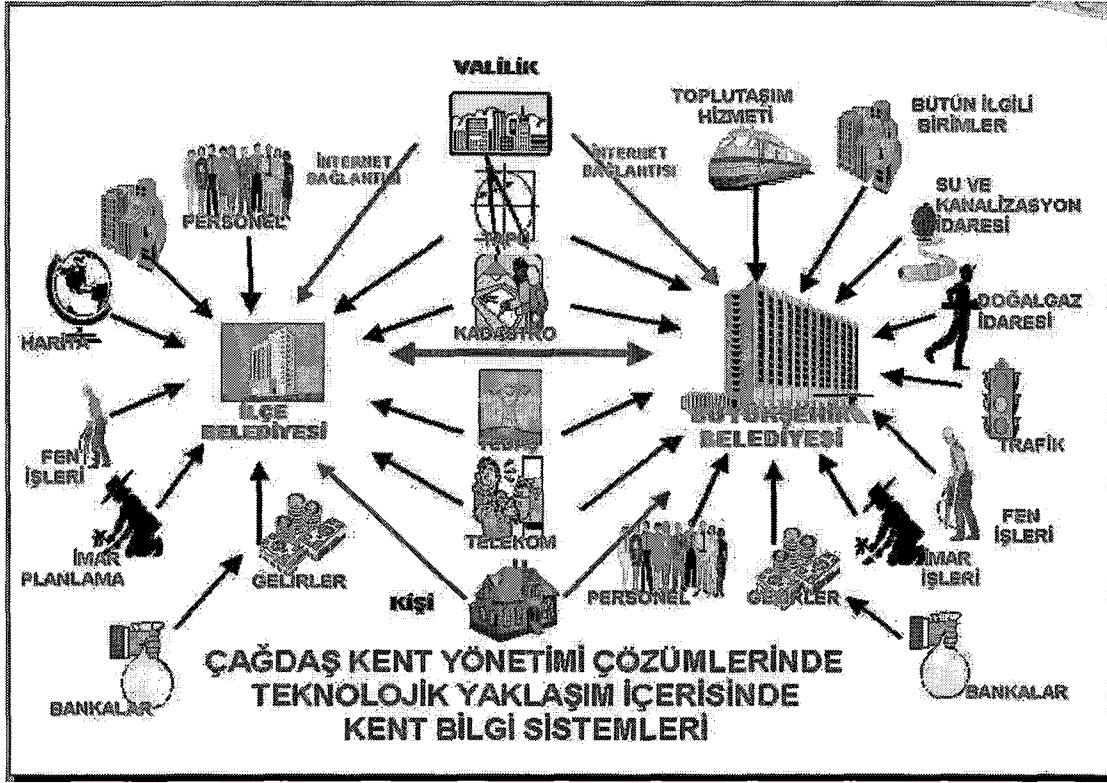
Sağlık İşleri Uygulamaları

Veteriner İşleri Uygulamaları

Satın Alma Uygulamaları

Kent Bilgi Sisteminin oluşturulması ile hizmet sağlayan kurumların ve hizmetten yararlananların ihtiyaçları daha çabuk, daha sağlıklı ve daha ekonomik olarak sağlanmış olacaktır.

Şekil 4.1’de kent bilgi sistemleri kullanımında ilişkili kurum ve kuruluşlar görülmektedir.



Şekil 4.1 Kent bilgi sistemleri kullanımında ilişkili olan kurum ve kuruluşlar [16]

4.2 KBS TEKNOLOJİSİNİN SAĞLADIKLARI

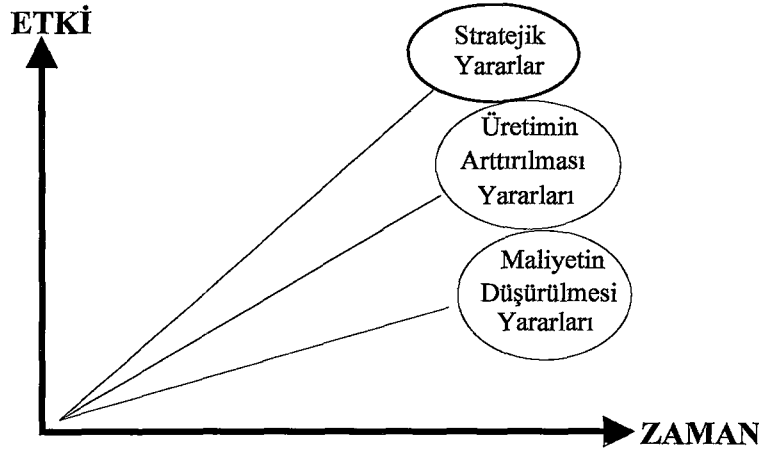
4.2.1 Maliyet / Yarar Analizi

“CBS’nin temel elemanları; donanım, yazılım, personel ve veridir. Bir CBS’nin uygulamaya konulması sırasında veri elde etmenin maliyeti yaklaşık olarak toplam maliyetin %80’i kadardır.” [17]

Mevcut yöntemlerle koordinasyonun tam anlamıyla sağlanamamış olması nedeniyle gelir kayıpları oluşmaktadır. Bu gelir kayıpları kaçak yapılaşmanın takip edilmesi, emlak, çevre temizlik verileri ile halihazır, tapu ve imar izin verileri karşılaştırılması ile vergi kaçakları, inşaat kaçakları tespit edilmesi, işyeri ruhsatı almamış olan işyerlerinin tespiti gibi konularda KBS olanakları ile gelir artışları sağlanacaktır. [16]

KBS ’nin kullanılması ile maliyet / yarar kriteri dikkate alınarak bir değerlendirme yapıldığında, sistemin kullanıcısı olan kurum ve kuruluşa, üretim artırma ve maliyeti düşürme yararları ile birlikte uzun vadede stratejik yararlar sağlayacağı görülür. [20]

Şekil 4.2’de KBS yararlarının zamanla etkisi görülmektedir.

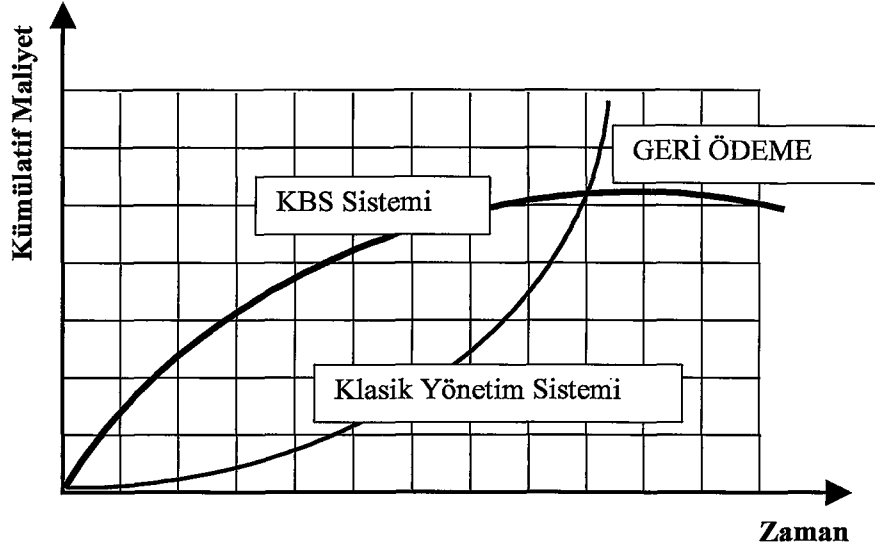


Şekil 4.2 Kurum ve kuruluşların yaşam sürecinde görülen KBS yararları [20]

Doğru verilerle hızlı bilgi edinmenin maliyeti değerlendirilmelidir. Yarar / maliyet analizleri genellikle yeni sistemin olumlu ve olumsuz yönlerinin eski klasik sistemlerle karşılaştırılması sonucu yapılır.

“Klasik yönetim sistemiyle otomatik yönetim sisteminin tipik bir kümülatif maliyet-zaman karşılaştırılmasının yapıldığı şekil 4.3 incelendiğinde, klasik yönetim sisteminde maliyetin zamanla arttığı gözlenmektedir. CBS yönteminde başlangıçtaki yüksek maliyetin nedeni, otomatik veri elde etme, teknoloji yatırımı, uzman personel alımı ve eğitimi ve sistemin sürekli bakımındır. Yatırımın azaldığı yerde CBS maliyetinin azaldığı ve sistemden yarar sağlanmaya başlandığı anlamına gelmektedir.” [7]

“Klasik yönetim sistemi ve KBS eğrisinin kesiştiği yer, maliyetin geri ödenmeye başladığı noktadır. Burada zaman için bir eğri verilmemiştir, zira zaman; yapılan yatırıma, alınan elemanlara, eğitime ve sistemin bakımına bağlı olarak uzayıp kısalabilir.” [20]



Şekil 4.3 Zaman içinde KBS ve klasik yönetim sistemlerinin kümülatif maliyetleri [20]

4.2.2 KBS Sistemlerinin Sağladığı Avantajlar

KBS kurmuş bir kurum, örneğin KBS sistemini oturtmuş bir belediye düşünüldüğünde eski yöntem ile yeni sistemin karşılaştırması daha net olarak görülecektir.

Nicelik bakımından:

- Kullanılan sistem içinde bilgisayar teknolojisinin yeri yoksa fazla insan gücü istihdamı, gereksiz bürokratik işlemler, fazla miktarda arşiv evrakları, gereksiz veri tekrarı olacaktır. Karşılaştırıldığında KBS'nin zaman, para ve insan gücü tasarrufu sağlayacağı açıktır.
- Kent yaşayan bir sistem olduğundan statik bir yapı olarak modellenemez. Sürekli bir değişime sahip olan kent verileri, sayısal olarak bulundurulduklarından güncelleştirmek çok daha kolay ve ucuzdur.
- Mevcut sistemlerde KBS verilerini güncelleştirmek, güncelleştirilen verileri ilgili haritalara işlemek, yeniden arşivlemek, ya da arşivlerde saklanan bu verilerle sözel verileri birleştirerek bir coğrafi analiz yapmak istendiğinde pek çok insanın, pek çok zaman harcayarak bu verileri bir araya getirmeleri, bilgi

üretmeleri ve buna göre karar vermeleri gerekmektedir ve bu oldukça güçtür. KBS'lerde ise verilere dayalı olarak istenilen bilgileri üretmek daha doğru ve daha hızlıdır.

- Uygun veri standardı ile bir başka CBS'den veri aktarmak, yeniden üretmekten daha ucuzdur.
- Kullanıcıya daha iyi karar alma imkanı verir.

Nitelik bakımından:

- Farklı yönetim birimleri, kurum ve kuruluşlar arasında bilgilerin paylaşımını sağlayarak birbirlerine ait bilgileri kullanmaya olanak sağlar.
- Bilgilerde hızlı değişim olması farklı yerlerde depolanan bilgiler arasında tutarsızlıklara yol açar. KBS ise bu tür karmaşayı ve tutarsızlıkları önler.
- KBS, farklı birim, kurum ve kuruluşların problemlere daha sistematik bir tarzda yaklaşmaları için gereken bütünsel etkinliğe teşvik eder.
- Bilgiler belli özelliklerine göre sınıflandırılabilirler. Bu sınıflandırmalar, çeşitli bilgilere ihtiyaç duyan birimlerin problemlerini çözmelerinde yardımcı olabilir.

4.3 MEVCUT KBS ÖRNEKLERİ

Ülkemizde pek yaygın olmayan kent bilgi sistemleri son senelerde gelişim göstermiş, pek çok belediye ve bazı özel kuruluşlar bu konuyla yakından ilgilenmeye başlamışlardır. Bu konudaki örnekler gittikçe çoğalmaktadır. Bursa ve Eskişehir belediyeleri şehir haritalarını ve pek çok coğrafi kent bilgisini, bilgisayarda bir CBS ortamına aktarmışlardır. Özel kuruluşların yaptığı örneklerden biri de Cihangir Güzelleştirme Derneği, Beyoğlu Belediyesi ve Sayısal Grafik'in ortak ve özel bir çalışması olan Cihangir Semt Bilgi Sistemidir. Yurtdışında ise, kent bilgi sistemleri çok yaygın olarak pek çok alanda kullanılmaktadır.

Yurtdışında pek çok karar aşamasında da KBS'lerden yararlanılır. Örneğin yeni açılacak bir fast food restoranı için en uygun yeri belirlemek amacıyla kullanılmaktadır. Hedef kitle gençler olarak kabul edilirse, liselerden 400m.'den daha uzak olmayan, ancak başka fast food restoranlarından en az 100m uzak olan alanlar sorgulanabilir, amaca yönelik başka kısıtlar da eklenerek sorgulamalar genişletilebilir ve KBS sisteminde görüntülenir. Böylece harita üzerinde gerekli şartları sağlayan alanlar belirlenerek konuyla ilgili iyi ve hızlı bir yaklaşım yapılmış olur.

KBS sistemlerinin özellikle yurtdışında çok yaygın olan bir diğer örneği de kampus bilgi sistemleridir. Fakülteler, enstitüler, yemek mekanları, spor alanları gibi pek çok bilgi içeren ve sorgulanabilen bu sistemler hem üniversite elemanları ve öğrencileri, hem de bu üniversitede okumak isteyen veya kampus ile ilgili bilgi edinmek isteyen yabancı kullanıcılar için kullanışlıdır.

Örnekler, kampus bilgi sistemlerinden, kapı numaralarına veya binalarda oturan insanların mesleklerine kadar bilgi içeren çok geniş veritabanlarına bağlı belediye KBS sistemlerinde kadar uzanır. Pek çok konuda öngörü sağlayan KBS'ler, karar destek sistemleri olarak kullanılırlar. Turistik amaçlı kent bilgi sistemleri, afet ve deprem öncesi ve sonrası çözüm sistemleri, askeri uygulamalar, polis, ambulans ve itfaiye için araç takip sistemleri ile bütünleştirilmiş CBS örnekleri gibi pek çok çeşit KBS bulunmaktadır.

4.4 BİR KBS PROJESİ OLUŞTURMA

4.4.1 Kullanılan Bilginin Yapısı

Kent Bilgi Sistemi, bir kente ait grafik ve grafik olmayan coğrafi bilgilerin ve bu bilgilere bağlı diğer verilerin, bir sistem içerisinde ele alınıp sınıflanması, düzenlenmesi, saklanması ve sistemdeki bu bilgilerin istenilen amaca uygun sorgulanarak analiz edilmesi yoluyla gerekli detay bilgiye ve buradan da sonuca en hızlı biçimde ulaşabilmenin bir yoludur.

Bu bilgi sisteminde veri tabanını oluşturan bilgilerin coğrafi bilgi olma özelliğini taşıması gerekmektedir. Coğrafi veriler, ormanlar, akarsular, binalar, yollar, köprüler, topografik, hidrografik, kadastral bilgiler, bitkiler, hayvanlar, nüfus dağılımları, sosyo-ekonomik bilgiler gibi verilerdir.

İnsanların barınma, eğitim, ulaşım vb. ortak ihtiyaçlarının etkin biçimde karşılanabilmesi için imar, planlama, altyapı vb. faaliyetlerinde öncelikle kentte bulunan doğal kaynaklar, mevcut arazi kullanımı, yapılar, tesisler, mülkiyet, planlama vb. güncel ve doğru olan verilerin kullanılması gereklidir.

Bir kuruluş, bir kent bilgi sistemi kurmadan önce detaylı bir ihtiyaç analizi yapmalıdır. Bilgiyi kimlerin hangi amaçla kullanacağı, kullanıcıların istediği hassasiyet, detay ve bilgilerin hangi formda istendiği incelenmelidir. Böyle bir ön hazırlık sürecinde, kullanıcının sadece bugünkü ihtiyaçları değil, ileriye dönük ihtiyaçları da göz önüne alınmalıdır. [17]

Kent Bilgi Sistemleri, kent düzeyinde veri içeren CBS'lerdir. Aşağıda, Intergraph firmasının GeoMedia yazılımı kullanılarak hazırlanmış bir KBS görüntüsü yer almaktadır. Kullanılan yazılımlar değişebilir ancak temel yapı ve fonksiyonlar değişmemektedir. Bir KBS 'de verinin genel gösterimi ve çalışma ortamı şekil 4.4'te gösterildiği gibi olacaktır.

Kent Bilgi Sistemi



4.4.2 Amaç Belirleme

KBS tasarlanmanın iki amacı olabilir. Birinci amaç, bir veya ilgili birkaç konuda analiz yapmaktır. Bu durumda analizin gerektirdiği seviyede ve detayda veri toplanarak KBS oluşturulur ve amaca yönelik olarak çalıştırılır. Bir diğer seçenek de, temel kent öğeleri ve bunlar ile ilgili genel verilerden oluşan bir KBS hazırlayarak kentlilerin hizmetine, genel fakat çok amaçlı kullanım için sunmaktır.

Özel amaçlı KBS 'lere örnek: Eğitim ve ilgili konularda kullanılmak üzere hazırlanan bir kent bilgi sisteminde kent haritası üzerinde gösterilecek ve analizlerde kullanılacak kent öğeleri okullar, dershaneler, çeşitli kurslar olabilir. Bu kent öğeleri belirlendikten sonra gerekecek özellikler toplanır. Bahsedilen KBS 'deki kent öğelerinden biri olan okullar için öznitelik bilgisi olarak toplam öğrenci sayısı, öğretmen sayısı, ve başka bir kent öğesi olan binalar için de eğitim düzeyi ya da iki lisan bilenlerin sayısı gibi bazı bilgiler toplanabilir. Daha sonra KBS sorgulama yetenekleri kullanılarak eğitim düzeyi dağılımı, yabancı lisanı olanlar ile lisan

kurslarının ilişkisi gibi bazı ilişkilendirmeler ve analizler yapılarak sonuç çıkarılabilir, ve sağlıklı yorumlar yapılabilir. Bu tür bir amaca yönelik olarak hazırlanmış bir KBS 'nin, kentte doğalgaz hatlarının geçtiği bölgeler içinde, itfaiye merkezlerine 2 km.'den uzak olmayan alanları bulma konusunda yardımcı olmayacağı açıktır. Önemli bir nokta, özel amaçlı KBS 'lerde performansın düşmemesi açısından gereksiz bilgilerden kaçınılmalıdır. Bu tür sistemler sadece ilgili bilgileri içermelidir.

Genel amaçlı KBS 'lerde ise, belli bir konuya yönelik olmayan bazı temel veriler toplanır. Bu KBS 'lerde kent öğeleri olarak okullar, karakollar, hastaneler, resmi kurumlar, park alanları, itfaiye merkezleri, metro hattı, telefon ve doğalgaz hatları gibi bazı temel kent öğeleri ile yer altı ve yerüstü ağ yapıları bulunacaktır. Bu kent öğeleri aslında bir çok örneği bulunan veri kümeleridir. Her veri kümesi ayrı bir katmanda saklanır ve her zaman tüm katmanlar açılmaz, gerekli katmanlar açılarak kullanılabilir.

4.4.3 Kent Öğelerini ve Özelliklerini Belirleme

CBS teknolojisinde 3 boyutlu sistemlere geçiş için yapılan çalışmalar sonucunda yakın gelecekte 3 boyutlu CBS kullanımı yaygınlaşacaktır, ancak günümüzde KBS, kentin bilgisayar ekranına taşınmış 2 boyutlu bir modeli olarak tanımlanabilir. Bu model kentin kendisi gibi dinamik bir yapıya sahiptir. Sürekli değişim gösteren, yenilenen kente paralel olarak, kentin bir modeli olan KBS'de devamlı değişim halindedir, devamlı güncellenen bir yapıdır.

Bir KBS hazırlanırken, kent hakkında edinilmek istenen bilgiler nelerdir belirlenir. Hazırlanan sistemin amacına göre bu bilgiler değişecektir. Kent hakkında yapılabilecek analizlerde, coğrafi boyutu ve/veya öznitelik değerleri bakımından sorgulanan, kent öğeleri nelerdir saptanmalıdır. Temel kent öğelerine örnek olarak parseller, binalar ve yollar verilebilir.

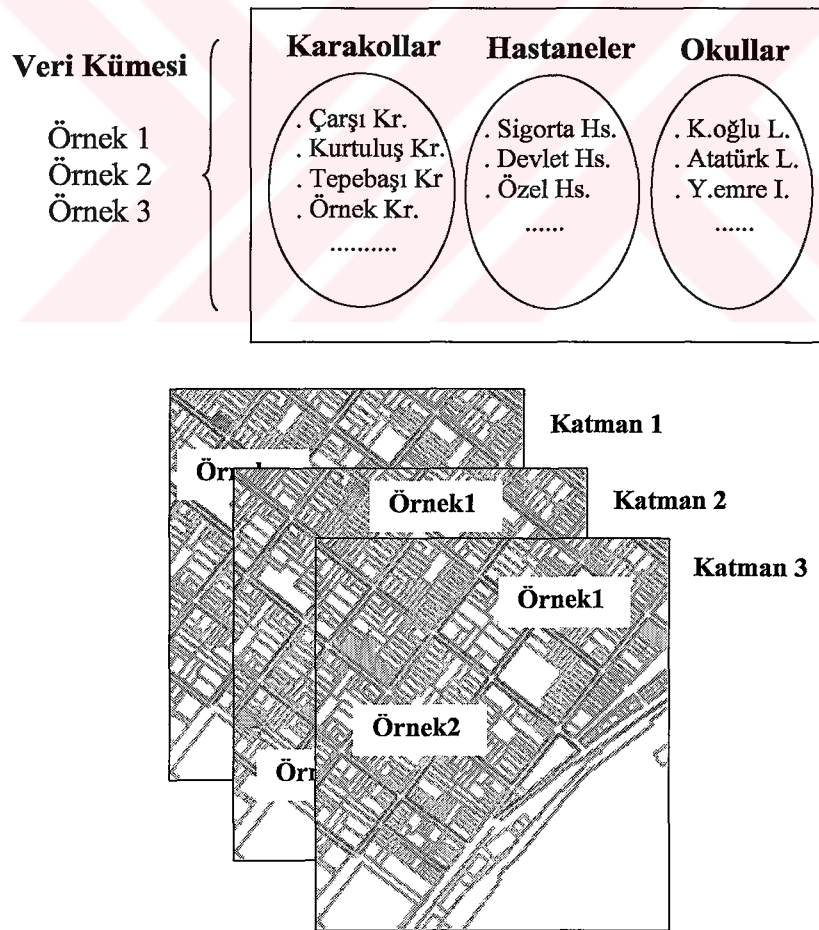
“Parseller”, parsel alanları ve parsel numaraları ile ilgili, “binalar”, ait oldukları mahalleler ve bina içinde barınan kişi sayısı ile ilgili, ve “yollar” da, uzunlukları ve cadde/sokak isimleri ile ilgili olarak sorgulanabilirler. Kent öğelerinin yukarıda bahsedildiği gibi en çok sorgulanabilecek, tahmin edilen ana özelliklerinde karar

kılınmalı ve kent öğelerinin öznitelik bilgisi olarak sisteme hangi verilerin girileceği belirlenmelidir.

4.4.4 Ayır Katmanlara Veri Girişi

Kent öğelerinden her biri için, bu veri kümesine ait tüm örneklerin çizildiği katmanlar oluşturulur. KBS sisteminde yer alacak kent öğelerinden biri olan okullar pek çok elemandan oluşan bir veri kümesidir. Bu veri kümesine ait tüm örnekler, çizim ortamında aynı katmana çizilirken, veritabanında da aynı tablonun farklı kayıtları olarak tutulacaklardır. Çoğu CBS yazılımında kullanıcı çizim penceresini açar ve örneğin okullar katmanını seçerek kent üzerindeki tüm okulları bu katmana çizer. Çizilen her poligon ayrı bir kayıt olarak düşünülecektir. Bu şekilde sıra ile tüm veri kümeleri için ayrı bir katman açılarak o veri kümesinin örnekleri çizilir.

Şekil 4.5'te veri kümeleri ve içerdikleri örneklerin katmanlar halinde saklanması görülmektedir.

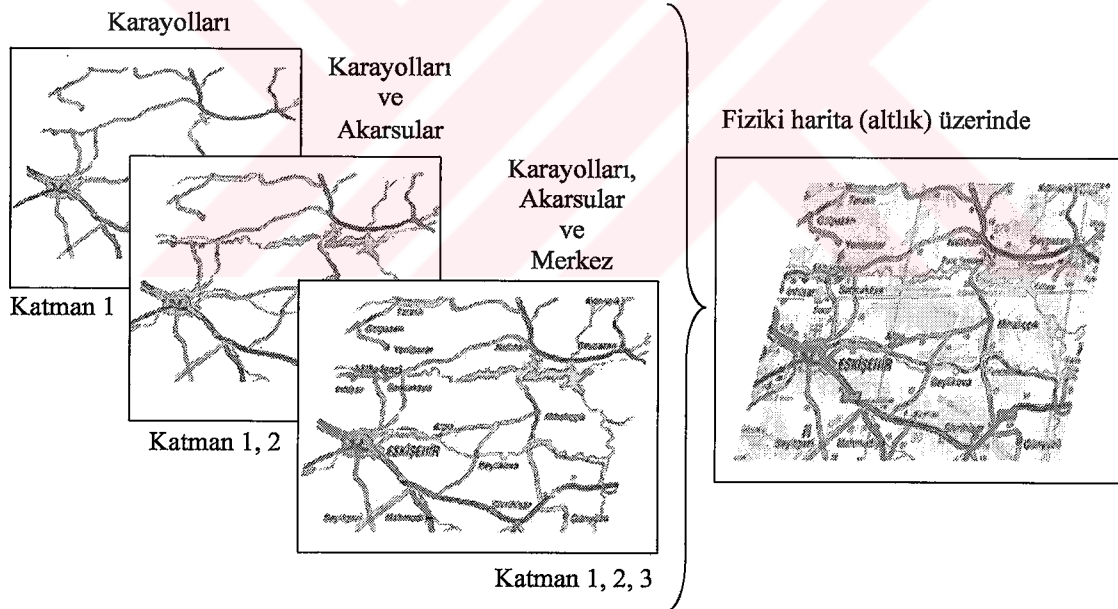


Şekil 4.5 Veri kümeleri ve içerdikleri örneklerin katman yapısında saklanması

Tüm katmanların gösteriminin belli bir altlık haritanın üzerine açılması isteniyorsa, öncelikle kente ait sade bir altlık kullanılabilir. KBS 'nin en sade seviyesini temsil etmek üzere hazırlanan altlık harita, belli bir veya birkaç kent ögesine ait veri kümelerini gösteren vektörel bir çizim veya bir uydu görüntüsü veya hava fotoğrafı gibi raster bir görüntü olabilir. Altlık harita kullanmak tasarımcının bir seçimidir. Hiç altlık kullanılmayan ve tüm veri katmanları boş bir sayfa üzerine açılan bir KBS de tasarlanabilir.

Kent Bilgi Sisteminin oluşturulmasında temel altlık haritalar olarak şu haritaların kullanıldığı gözlenmiştir: Halihazır haritalar, imar planları, kadastral haritalar, kamulaştırma haritaları, altyapı haritaları, doğalgaz, su ve kanalizasyon, telefon hatları, elektrik şebekesi, kablolu yayın, petrol boru hatları. [16]

Şekil 4.6'da karayolları, akarsular ve merkez isimlerini saklayan bir, iki ve üçüncü katmanların sırayla üst üste açılması görülmektedir. En son olarak ise altlık katman da eklenerek görüntü tamamlanmıştır.

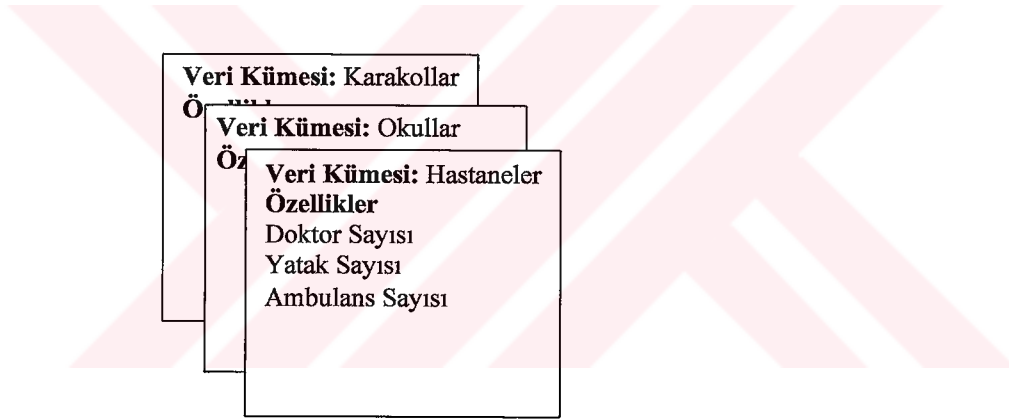


Şekil 4.6 Katmanların üst üste açılması ve altlık harita kullanılması

Bir CBS yazılımında farklı veri kümelerine ait olan elemanlar farklı katmanlara çizilirler. Bu çizim esnasında her eleman çizimi tamamlandıktan sonra açılan bir pencere yardımı ile veya daha sonra veritabanına ulaşarak, her elemanın belirlenen

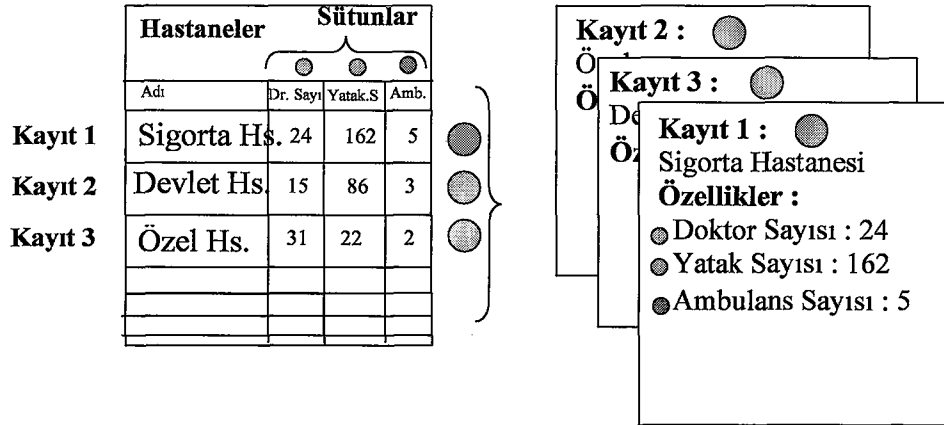
özellikleri ile ilgili bilgiler sisteme girilir. Bu şekilde, arka planda veritabanı bilgileri doldurulmuş olmaktadır.

Bir KBS sistemi, konuya ilişkin mümkün olan en geniş veri kümesini kapsamak üzere tasarlanmalıdır, ya da yapılacak analizlerde her değişkenin değerlendirilmesi ve ilişkilendirilmesi için ayrı katmanlarda kayıtlı çok veri kümesine sahip olmalıdır. Veri kümelerinin ve katmanlarının çok olması kullanım olanaklarını hem arttırmakta hem de daha doğru sonuçlara ulaşılmasını ve çok yönlü araştırmaların yapılabilmesi imkanını sağlamaktadır. Aslında altlık üstüne konabilecek ne kadar çok veri kümesi tanımlanabiliyorsa, o kadar kullanım alanı ortaya çıkar. Bir KBS 'nin veri bakımından zengin olması ondan çok daha fazla yararlanılmasını sağlayacaktır. Grafikselsel olarak görülen katman yapısı, veritabanındaki tablo yapısına denk gelmektedir. Şekil 4.7'de, KBS sistemindeki katmanların veritabanındaki tablo olan karşılıkları görülmektedir.



Şekil 4.7 Veritabanı tarafındaki katman yapısının sembolize edilmesi

Bir KBS için toplanan verilerden her birinin veri tabanında bir kayıt olarak saklandığı, ve bu örneklere ait özelliklerin de bu bilgiyi göstermek üzere belirlenen sütunlara kaydedildiği şekil 4.8'de görülmektedir. Her bir kayıt, bir satıra her bir özellik de bir sütuna denk gelmektedir.



Şekil 4.8 Veritabanında veri kümesi örneklerinin satır, özelliklerinin ise sütunları oluşturması

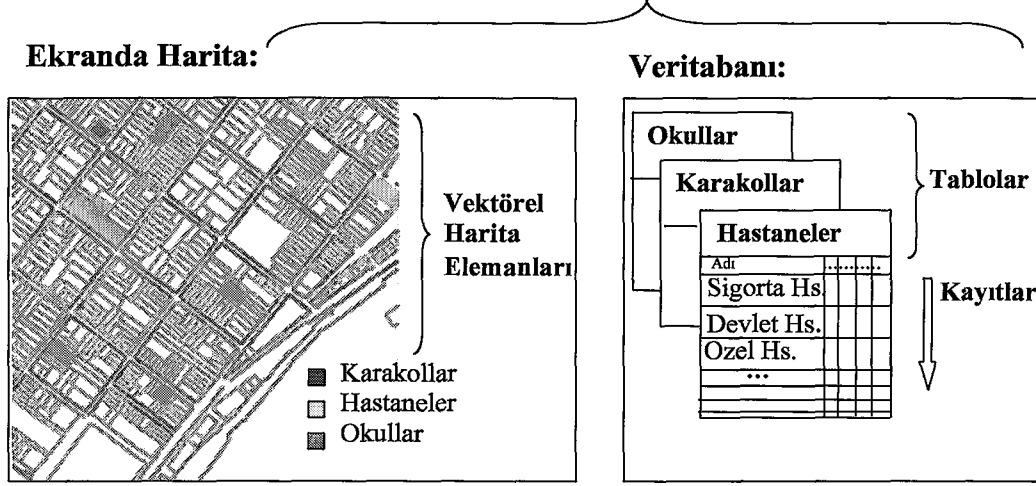
Katmanların her birine atılan veri kümeleri ile ilgili mümkün olduğunca çok özelliğin veritabanına işlenmesi de aynı şekilde yararlı olacaktır. Örneğin bir binanın katsayısı, bina türü, bina içerisinde barınan kişi sayısı, deprem konusu ile ilgili bir analiz için kullanılacak gerekli verilerdir. Bu binalarda yaşayan çocuk sayısı ve beslenen evcil hayvanlar ile ilgili olarak toplanacak veri ise bu ikisi arasındaki ilişkiyi saptamaya çalışan psikoloji konulu bir araştırma için yararlı olacaktır.

Sonuç olarak, hem her uygulamada tüm veri kümelerine ihtiyaç duyulmayabilir, hem de veri kümeleri ile ilgili toplanan tüm bilgiler bir analiz için kullanılmayabilir.

4.4.5 Sistemin Veritabanı Tarafı

Kent Bilgi Sistemleri, ekrandaki coğrafi elemanların arka planda veritabanında kayıtlı pek çok bilgiyle ilişkili olduğu harita üretim sistemleridir. KBS sisteminde, veri kümelerine ait her katman veri tabanında başka bir tablodur, ekrandaki her eleman da tablolardaki başka bir kayıta denk gelir. KBS, sistemdeki tüm harita katmanları ve bunların ilişkili olduğu tabloların tümüdür. Başka bir deyişle, harita elemanları ve bunları öznitelik bilgileri ile birlikte saklayan veritabanının tümüdür. Şekil 4.9'da kent bilgi sisteminin, hem ekrandaki harita, hem de veritabanından oluşan yapısı gösterilmiştir.

Kent Bilgi Sistemi



Şekil 4.9 KBS, hem grafik veriler hem de bunlarla ilişkili veritabanının bütünüdür.

Kent Bilgi Sistemlerinde, görüntülenen harita elemanları aynı zamanda etkileşimli olarak tasarlanabilir. Örneğin her elemanın adının harita üzerinde yazılı olarak bulunması bir karışıklığa sebep olacak ise, nesne isimlerinin fare üzerlerine gelince ekranda görüntülenmesi daha kullanışlı olacaktır.

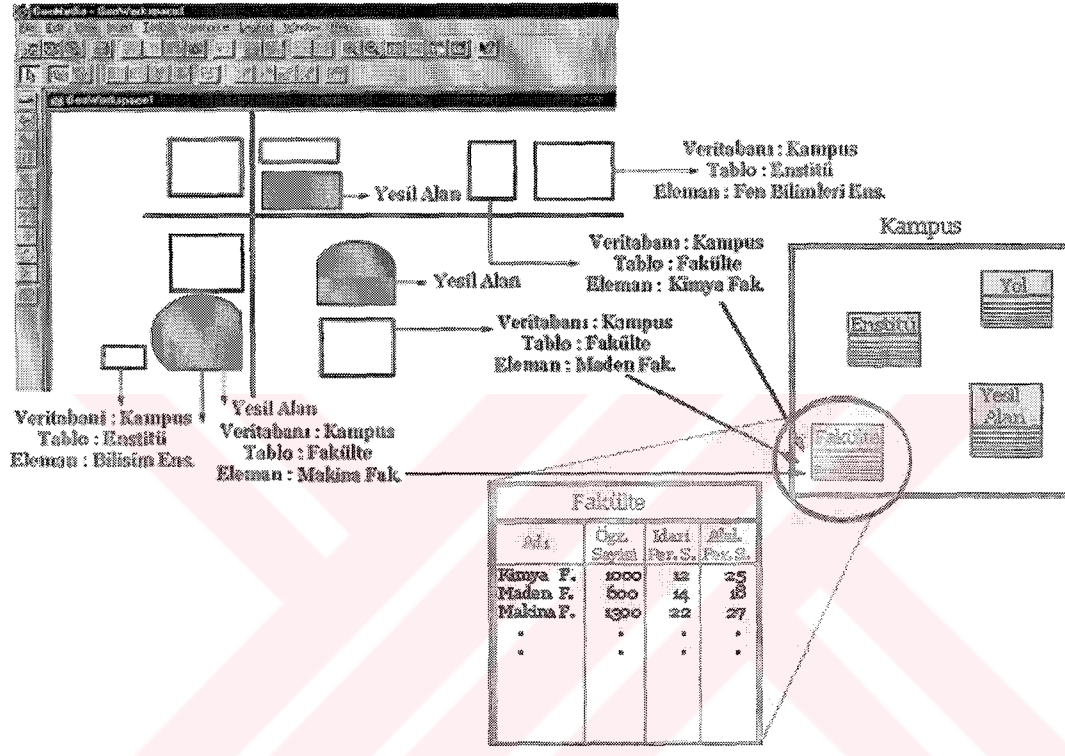
Kullanıcı, hangi veri kümesine ait eleman çizecek ise belirtir ve program da çizilen elemanları görüntüler. Bu sırada, kullanıcı veritabanı ile ilgili hiçbir işlem yapmamasına rağmen, veritabanında olan gelişmeler şunlardır: Veritabanında, tanımlanan her veri kümesi için ayrı bir tablo açılır. Çizilen her eleman, örneği olduğu veri kümesine ait olan tabloya bir kayıt olarak eklenir. Bu tabloya, çizilen elemanın her noktasına ait koordinatlar da otomatik olarak girilerek, elemanın grafik özellikleri de kaydedilmiş olur.

Tüm bunları basit bir kampus bilgi sistemi için toplarsak:

İhtiyaç duyulabilecek veri kümelerinden bazıları, “fakülte”, “enstitü”, “büfe”, “laboratuvar”, “yol”, “yeşil alan” olabilir.

Örneğin, “Uçak ve Uzay Bilimleri Fakültesi”, “İnşaat Fakültesi”, “Maden Fakültesi”, “fakülte” veri kümesinin elemanlarıdır.

Bu “fakülte” örneklerinin her biri çizilirken “fakülte” tablosunun ayrı bir kaydı olarak tutulurlar. Bir CBS yazılımında, kullanıcı veritabanına girmeden, henüz çizim ortamında çalışır iken, bir eleman çizer ve tamamladıktan sonra bu grafik elemana tıklanırsa o elemana ait bir pencere açılır. Bu açılan pencereden elemana ait öznelik bilgileri girilebilir. Bu bilgiler de otomatik olarak veritabanında o elemanın satırına kaydedilir. Şekil 4.10’da grafik elemanlar ve veritabanı ilişkisi yer almaktadır.



Şekil 4.10 Örnek bir kampus bilgi sistemi için grafik elemanlar ve veritabanı ilişkisi

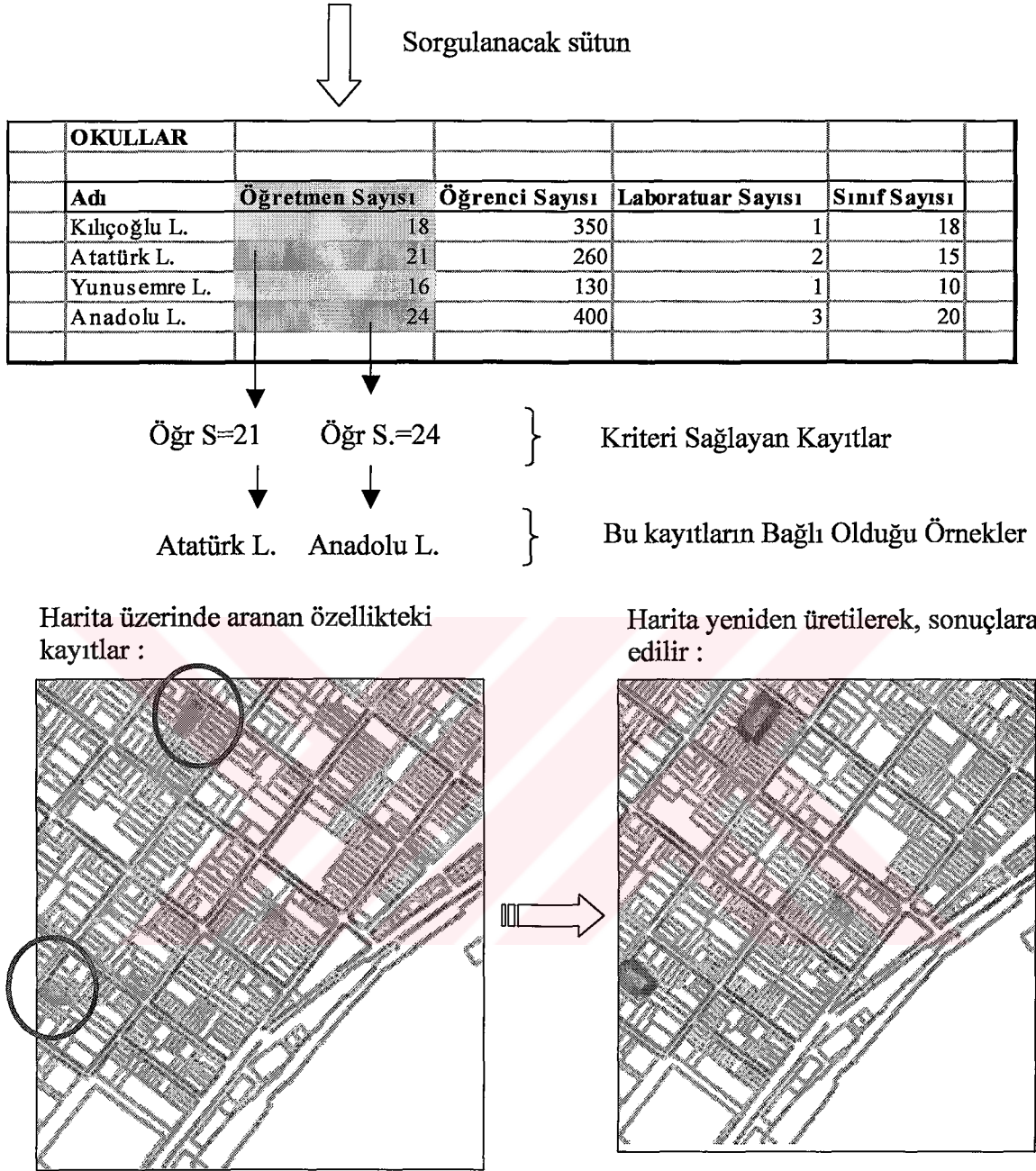
4.4.6 Sorgulama

KBS Sistemlerinin en faydalı özelliği, kent hakkında sorgulama imkanı tanınmasıdır. Yazılımın kullandığı arayüz aracılığı ile, değişik sorgulamalar yapılabilir. Basit bir sorgulama bir isim girilerek buna ait çizimin harita üzerinde işaretli olarak gösterilmesi olabilir. Bir cadde veya bir hastane adı girilerek nerede olduklarının gösterilmesi istenebilir.

KBS 'ler ile daha karmaşık sorgulamalar yapmak da mümkündür. Örneğin uzunluğu 500m'yi geçen caddeler hangileridir ya da yatak sayısı 50'den fazla olan hastaneler hangileridir gibi sorgulamaların cevabı bir örnekler kümesi olacaktır.

Sorgu aşamasında işlemler şu şekilde yürür: Sistem veritabanında ilgili tabloyu bulur, aranan özellik bilgisinin girildiği sütunu bulur ve bu sütunu istenen kritere uyan kayıtları bulmak üzere taramaya başlar. Arzu edilen niteliğe sahip olan kayıtları bulur, daha sonra bu kayıtlar hangi elemanlara ait ise onlar sorgu sonucunun gösterimi için tanımlanan kurala göre farklı gösterilir. Sonuç veya sonuçlar, veri kümesindeki diğer elemanlardan farklı bir renkle veya nesnenin içi renk ile doldurulmuş olarak ya da sınırları çok daha kalın çizilmiş olarak görüntülenebilir. Yani, harita üzerinde yapılan sorgunun sonucu kullanıcıya işaret edilir. Örneğin yukarıdaki tipte bir sorgulama ele alınsın: “Öğretmen sayısı 20’den fazla olan liseler hangileridir?” Bu durumda, sistem sonuca ulaşmak için ilk önce ilgili veri kümesinin veritabanındaki karşılığı olan tabloyu arayacaktır. Burada ilgili tablo, “liseler” tablosudur. Daha sonra kriter olacak öznelik bilgisinin hangi sütun olduğu bulunur. Yani, “öğretmen sayısı” sütunu bulunur. Bu sütundaki kayıtlar taranarak, aranan değerle eşleşenler ve eşleşmeyenler olarak ayrılabilirler. Bu tarama sonucunda öğretmen sayısı 20’den fazla olan liseler listesi çıkarılmıştır. Harita tekrar üretilir. Aranan özellikteki elemanlar, harita üzerinde farklı bir şekilde gösterilerek sonuca işaret edilir.

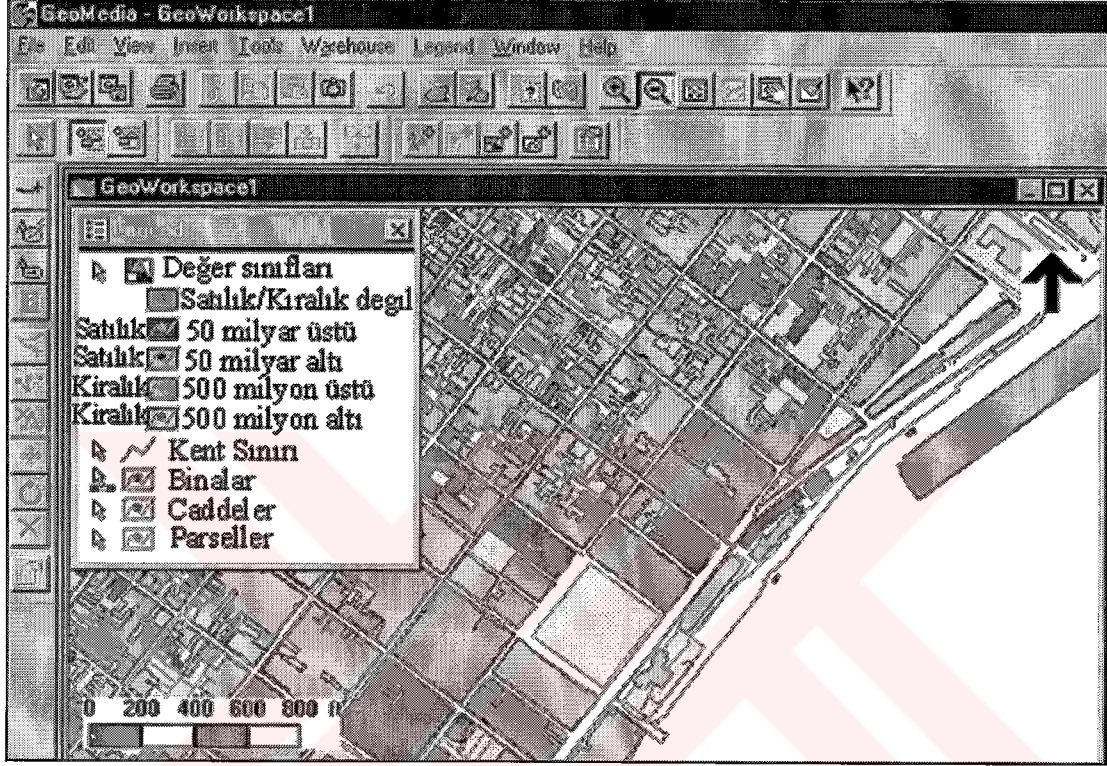
Şekil 4.11’de bahsedilen sorgulama sürecinde, veritabanı üzerinde tarama yapılarak, yeniden üretilecek haritada farklı gösterilecek olan elemanların belirlenmesi gösterilmiştir.



Şekil 4.11 Bir sorgulama sürecinde veritabanının taranarak sonuç listesinin çıkarılması ve yeni harita üretilmesi

Bir diğer sorgulama türü ise, veritabanına ulaşarak bir özelliğe göre sınıflandırma yapmaktır. Bu durumda ilgilenilen özelliğe ait sütun taranırken belli bir değer aranmamakta, bu sütun değerlere göre sınıflandırılmaktadır. Daha sonra da bu sınıflar için farklı gösterimler yapılır ve yeniden üretilen harita üzerinde gösterilir. Örnek olarak bir emlakçının kullandığı bir KBS uygulamasını ele alalım. Kentte

emlakçıya verilmiş binalar, satılık olanlar, kiralık olanlar ve satılık veya kiralık olmayanlar şeklinde gruplandırılabilir. Daha ayrıntılı bir sınıflandırma ile binalar, kira ve satış değerlerine göre belli aralıklar verilerek ayrılabilir. Şekil 4.12’deki sistem, binaların değerinin kaydedildiği sütunu tararken kira ve satış değeri ile ilgili ikişer sınıf oluşturacak şekilde sınıflandırma yaparak ve bunları farklı şekilde göstermiştir.

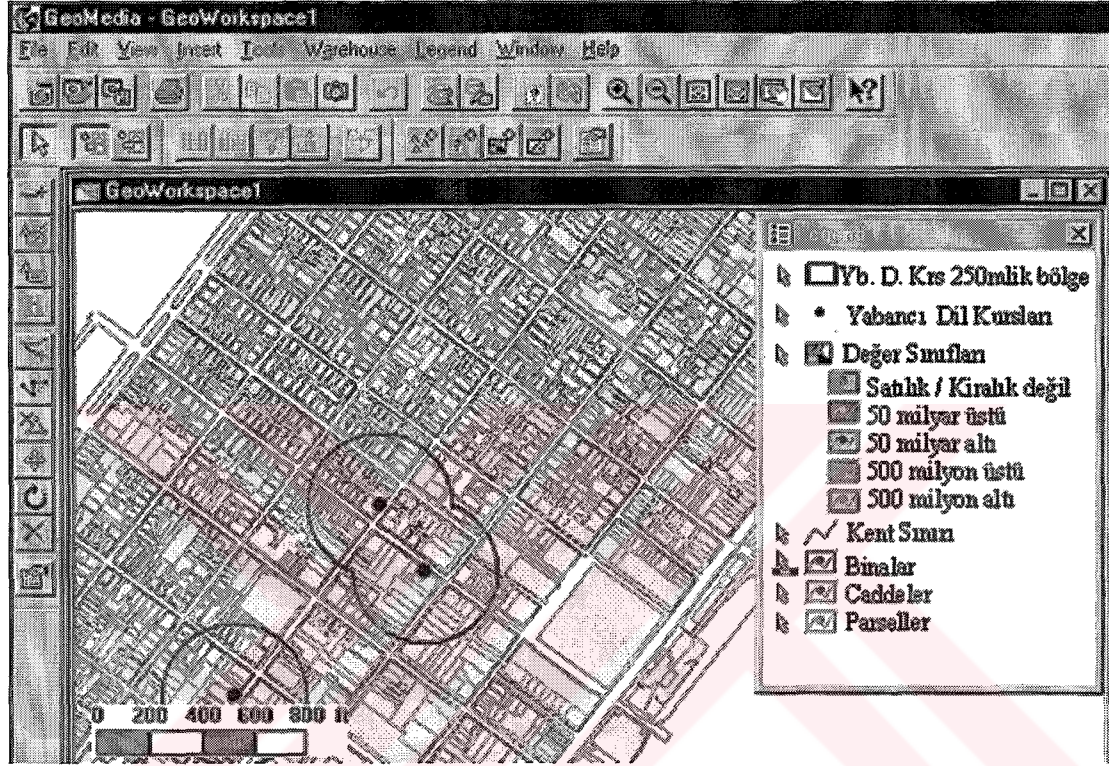


Şekil 4.12 Veritabanındaki değerlerin sınıflandırılarak gösterimi

Veri özellikleri girilirken, sorgulamalarda ne derece detaylı bilgi aranacak ise o detayda bilgi girilmelidir. Bu şu demektir: Örneğin, emlakçının kullandığı, bahsedilen bilgi sisteminde, satılık evlerin değerleri için ucuz, normal ve pahalı olarak 3 farklı sınıf tanımlanabilir. Bu durumda veritabanında satılık evler ile ilgili tablonun değer sütununa, her eleman için ya ucuz ya normal ya da pahalı yazılacaktır ve bir sorgulama ile ancak “satılık evlerden pahalı olanlar hangileridir?” şeklinde bir soru cevaplanabilecektir. Bu göreceli bir kavramdır ve pek çok kişiye net bir fikir veremeyecektir. Örneğin, “20 milyar ile 30 milyar arası olan evler hangileridir?” şeklinde bir soru yönelten müşteriye net bir cevap vermek mümkün olmayacaktır.

Bu durum göz önüne alınarak, veritabanına bu tür bir sorunun cevabını ararken kullanılabilecek uygun bilgi girilmelidir.

Bir başka sorgulama ise, şehirdeki dil kurslarının 250m. yakınındaki bölgeleri harita üzerinde göstermek üzere yapılabilir. Bu durumda dil kursları etrafına yarıçapı 250 m. olan bir çember çizilerek bu bölgeler işaretlenmiş olacaktır. Şekil 4.13’de yabancı dil kursları ve bunların 250m.’lik yakın çevre gösterilmiştir.

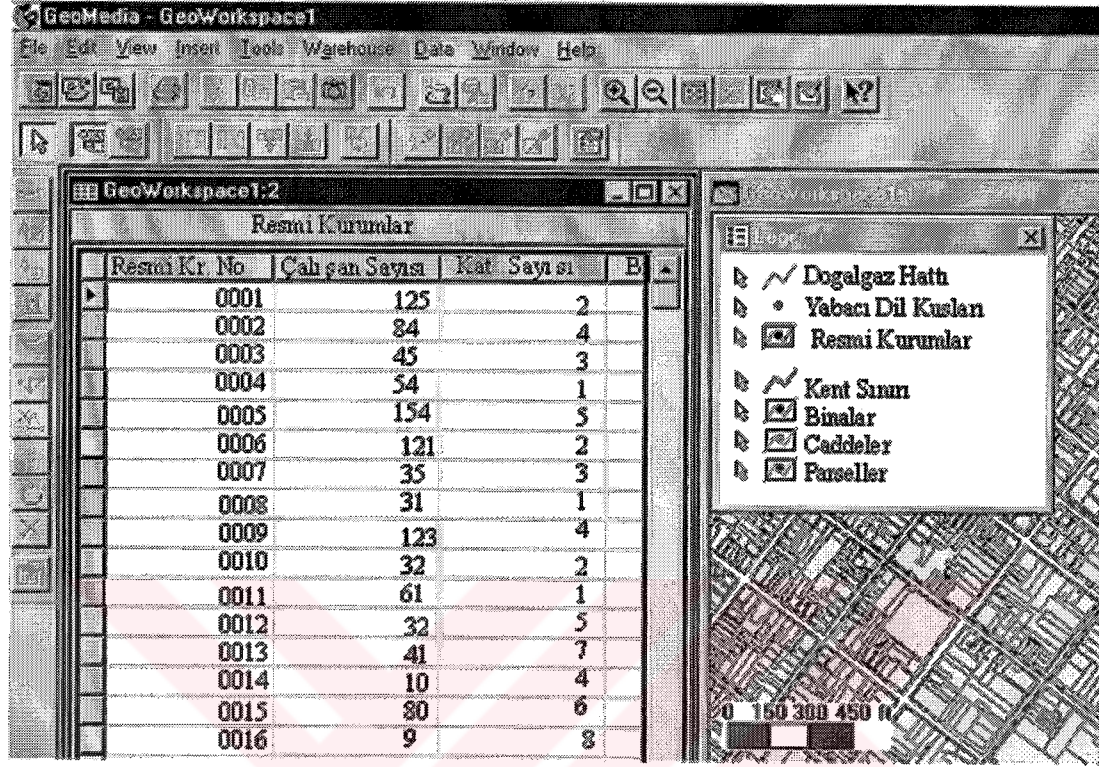


Şekil 4.13 Bir veri grubuna belli uzaklıktaki çevre bölgenin gösterimi

Daha sonra sadece bu bölgeler içerisindeki elemanları ilgilendiren bazı sorgulamalar yapılabilir: “Dil kurslarına 250m.’den uzak olmayan bölgelerde yabancı dilde yayın satan kitapçılar kaç tanedir, hangileridir, ya da bu tür yayınlardan elde ettikleri satış karı ne kadardır? ...” gibi.

Bu tür devam edecek sorgulamalarda görüldüğü gibi bir diğer ihtiyaç, sorgu sonuçlarını harita üzerinde olduğu gibi, veri penceresinde de görüntülemek, sonuçları liste olarak almak olabilir. Sonuçlar hem coğrafi konumları görmek açısından harita penceresi ile hem de sonuçta çıkan elemanları ve bu elemanların veritabanındaki diğer özelliklerini görebilmek ve listelemek için veri penceresi ile

gösterilir. Şekil 4.14'te resmi kurumların sağ tarafta harita üzerinde, sol tarafta ise veri penceresinde gösterildiği bir KBS ortamı görülmektedir.



Şekil 4.14 Harita penceresi ile veri penceresinin aynı anda izlenebilmesi

4.5 KBS ile Tümlleştirilmiş Sistemler

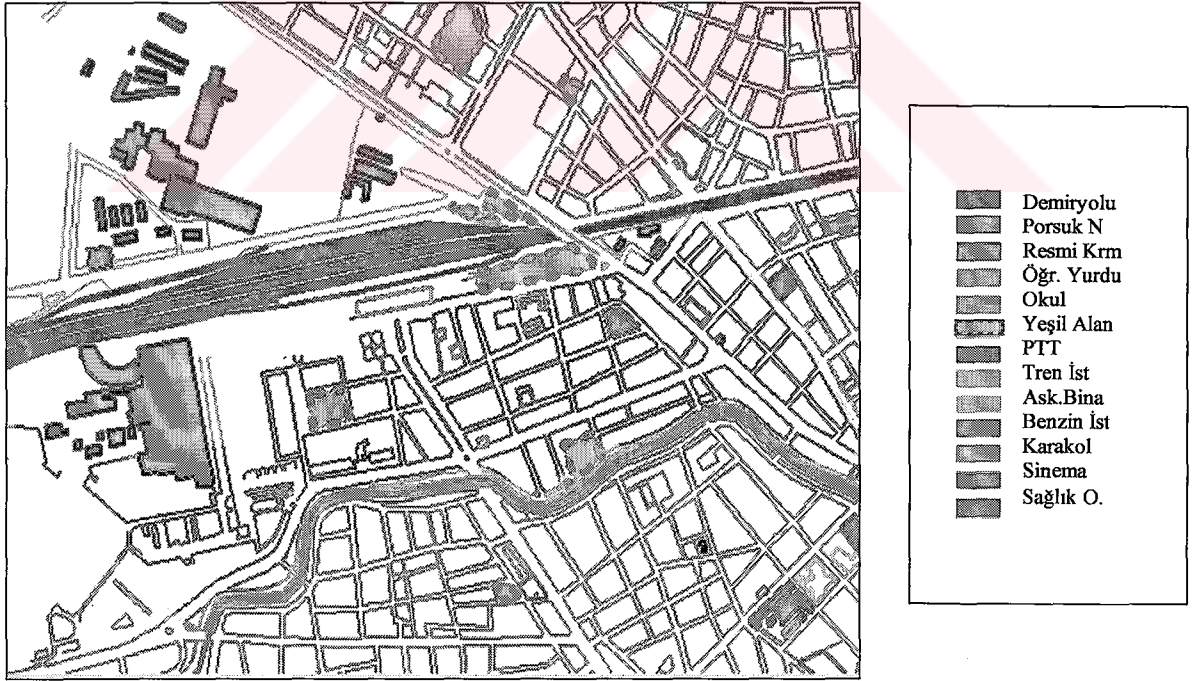
KBS Sistemlerinin daha gelişmiş bazı örneklerinde canlı veri girişi olanağı vardır. Özellikle araç takip sistemlerinde, araca takılan GPS ile KBS sistemi arasında kurulan bağlantı aracın konumunun devamlı olarak sisteme iletilmesini sağlar. Bu devamlı veri girişi, bilgisayar ekranından araç takibinin canlı olarak yapılabilmesine yarar. Özellikle ambulansları yönlendirmede kullanımı çok faydalıdır. Trafik yoğunluğu, tek yön uygulaması olan cadde ve sokaklar gibi bazı parametreler sisteme kısıt olarak tanıtıldıktan sonra, harita üzerinde çok daha rahat olarak aracın mevcut durumu ve en iyi hareket olanağı hakkında karar verilebilir, sürücü ile telsiz iletişimi kurularak sürücüyü doğru yönlendirmek mümkün olur.

Daha da gelişmiş sistemlerde, belirtilen kısıtları sağlayarak, verilen iki nokta arasındaki en kısa yolu bulan yazılım bileşenleri vardır.

KBS Sistemlerinin tümleştirilebileceği teknolojiler gün geçtikçe artmaktadır. Bunlara örnek olarak modelleme, internet teknolojileri ve yapay zeka (AI) verilebilir. Modelleme teknolojisi ile CBS teknolojilerinin birleştirilmesi, ortamı 3 boyutlu CBS sistemlerine dönüştürür.

İnternet teknolojileri ile KBS 'nin birleştirilmesinde gaye, KBS 'yi çok kullanıcıya açmak ve KBS 'nin asıl amacına ulaşmaktır. KBS ile yapay zeka ve öğrenen sistemler ise yeni teknoloji ürünü bazı taşıtlarda tümleştirilmiştir. Bu taşıtlar kendilerinde kayıtlı bulundurdıkları KBS sistemine, yolculuk esnasında öğrendikleri yol bilgilerini de ekleyerek çevrelerini tanımaktadırlar.

Şekil 4.15'te, Eskişehir'e ilişkin Kent Bilgi Sistemi görülmektedir. Genel amaçlı hazırlanan bu kent bilgi sisteminde, internet üzerinden bağlanan kullanıcı yan tarafta görülen kent öğelerinin yerleri hakkında bilgi edinmektedir.



Şekil 4.15 Eskişehir kent bilgi sisteminden bir görüntü

5. KENT BİLGİ SİSTEMLERİNİN İNTERNET TEKNOLOJİLERİ İLE TÜMLEŞTİRİLMESİ

5.1 İNTERNET TEKNOLOJİLERİ

5.1.1 İnternetin Tanımı ve Tarihçesi

İnternet, birbirine telefon hatları ile bağlı, ortak bir dili konuşan ya da ortak bir protokola TCP/IP (Transmission Control Protocol / Internet Protocol)'ye sahip olan çok sayıda bilgisayarın oluşturduğu uluslararası bir ağıdır. Aslında bu ağın bir üniversitedeki veya kuruluştaki Unix ağı, ya da iki bilgisayarın birbirine bağlanmasından bir farkı yoktur.

İki bilgisayar ortak bir protokola sahip oldukları, aynı dili konuştukları sürece, bilgi alışverişi yapabilirler, ve internet de aslında tam anlamı ile bu olay ile ilgilidir –yani bilgi alışverişi ve paylaşımı. Asıl görevi ise iletim ve iletişimdir. [21]

“İnternet 1969 yılında Amerika Birleşik Devletleri Savunma Bakanlığı'na (U.S. Department of Defense) bağlı olarak çalışan İleri Araştırma Projeleri Ajansı (Advanced Research Projects Agency – ARPA) tarafından deneysel amaçlı bir ağ olarak faaliyete geçti. Bu ağ (o zamanki adı ile ARPANET) araştırmacı bilim adamlarının birbirleri ile haberleşmesi amacıyla dizayn edilmişti. ARPANET ilk olarak dört bilgisayardan oluşmaktaydı. ARPANET'i kontrol için güvenlik açısından ne bir merkezi bilgisayar ne de bir merkezi otorite yoktu. Ancak 1972 yılı itibari ile, 50 üniversite ve askeri araştırma sitesinin ARPANET erişimi mevcuttu. Sonuç olarak ARPANET bilgisayarlar arasında birçok yol kullanabilecek şekilde tasarlandı, böylece bir mesaj tek bir sabit yol ile değil mümkün olan herhangi bir yol ile hedefine ulaşabilecekti. Yani bir bilgisayar çalışmaz hale geldiğinde, diğerleri onun etrafından dolaşarak birbirleri ile haberleşmeye devam edebileceklerdi.” [22]

Farklı bilgisayarların birbirlerine bağlanması neticesinde, dizayn edenler bu farklı bilgisayarların birbirleri ile konuşabilecekleri ortak bir dil geliştirmek zorunda kaldılar. Onların çözümü bir iletişim protokolü geliştirmek oldu, ve bu protokol TCP/IP olarak bilinmeye başladı. TCP/IP 1983 yılında internette kullanılan standart protokol olarak yerleşti ve halen de standart olma özelliğini korumaktadır. 1980'lerde bazı ağlar (Amerikan Ulusal Bilim Vakfı'nın beş süper bilgisayar birbirine bağlayan ağ dahil) ortaya çıktı. Zamanla, bütün bu halka açık ve özel ağlar birbirine bağlandı ve böylece bu ağlarda bulunan herhangi bir bilgisayar bütün ağlara ulaşabilir hale geldi. Günümüzde internet, Amerika Birleşik Devletleri başta olmak üzere 160'dan fazla ülkenin akademik, askeri, yönetim, ve ticari kurumlarını bir araya getirmektedir. [22]

5.1.2 Internet Kullanımı

Günümüzde internet kullanılarak, kaynak taraması, uzaktan eğitim, e-posta iletişimi, dosya aktarımı, kişisel ve kuruluş sayfalarına erişim, sohbet, alışveriş, canlı tv yayıncılığı, reklamcılık gibi pek çok iş yapılmaktadır. Gelecekte ise, günlük hayattaki olayların çok daha büyük bir kısmı internet üzerinden yapılacaktır.

Ancak internette yapılabilecek temel aktiviteler olarak şu işlemler sayılacaktır: İletişim, doküman ve dosya aktarımı, belirli konulara sahip ilan tahtalarından yararlanma, etkileşimli arama tarama. Bu aktiviteleri destekleyen pek çok araç ve teknoloji vardır: E-mail, ftp, telnet, usenet, archie, gopher, veronica, WAIS, WWW gibi. [21]

Internet kullanmak için gereken donanım bir bilgisayar ve internet erişimi sağlayacak çok kanallı bir bağlantıya dahil olmak veya bir modemdir. Kullanıcı ile eriştiği bilgisayar arasındaki temel araç olan telekomünikasyonu sağlayacak yazılıma ek olarak, bir de internet üzerinde e-mail gönderme, ftp işlemleri, telnete bağlanma gibi bazı aktiviteler için gereken yazılımlar olacaktır.

5.1.3 FTP Aracı, WWW Standardı ve Web Sayfaları

FTP (File Transfer Protocol) : Bir dosyayı bir yerden başka bir yere aktarmayı sağlayan internet aracıdır. Burada ‘dosya’ ile kastedilen metinler, resimler, görüntüler, hareketli görüntüler, sesler ve yazılımlar birçok dijital varlık türü olabilir. FTP ile dosya aktarımının en karakteristik özelliği farklı bilgisayarlar arasında da bunu gerçekleştirebilmesidir. Örneğin bir PC’de çalışırken, Unix kullanan bir bilgisayarda depolanmış bir dosya da çekilebilir.

WWW (World Wide Web) : Internette birbirinden farklı bilgisayarlara dağılmış multimedya verilerine daha kolay ulaşabilmek için 1992 yılında geliştirilmiş bir standarttır. [22] Web, CERN’de (European Laboratory For Practice Physics)’de geliştirilmiştir. İlk amacı, isteyen kişilerin bir ağ bağlantısı içinde, ortak ve basit bir protokol aracılığı ile uzak bir sistemde bulunan okunabilir bilgiye ulaşmalarını sağlamaktır. Bu durumda pek çok bilim adamı yazı, grafik, şekil, veritabanı alışverişinde bulunabilecekti. Ayrıca bu bilgiler birbirlerine ‘hypertext’ kavramı ile birleştirilebilecekti. [21]

‘Hypertext’, web kullanımını en etkili yapan özelliktir. Bir hypertextin arkasında yatan temel kavram, bilginin saklanıp hiyerarşik olmayan bir yapıda kazanılabilmesidir. Yani bilgiye klasörler arası ilerleyerek ulaşmanın dışında, bir yerden başka bir yere linkler aracılığı ile de atlanabilir.

Web sayfaları, internet üzerinde belli bir adrese sahip, bilgi ve görsel içerik sunan, ses teknolojisi ile desteklenebilen, diğer web sayfalarına bağlanmak üzere link bulundurabilen, arama yapılabilen, güncellenen, ziyaretçisinden bilgi girişi kabul edebilen, etkileşimli sayfalardır.

5.1.4 HTTP Protokolü ve URL Adresi

HTTP (Hypertext Transfer Protokol) Protokolü: WWW’de olanların hepsi HTTP internet protokolü kullanımı ile gerçekleşir. HTTP internete bağlı bilgisayarlar arasında dosya alışverişlerine olanak sağlayan kuralların oluşturduğu standarda verilen isimdir. HTTP’nin konuştuğu temel dil HTML (Hypertext Markup Language)’dir.

Bir web sunucusunun içinde HTML ve diğer multimedya dosyalarından başka HTTP istemlerini bekleyen ve onları düzenleyen bir yazılım bulunur. Bu yazılıma ‘HTTP Daemon’ adı verilir. Kullanılan tarayıcı bir HTTP istemcisidir ve bir linke geçiş yapıldığında bir http isteminde bulunulur. Bunun üzerine sunucudaki HTTP Daemon bu istemi alır ve gereken dosyayı (HTML) yükler. [22]

URL (Uniform Resource Locators) : Internette bir şeyin yerini belirlemenin en temel yoludur. Bu yer (adres) bir bilgisayarın adı olabilecek kadar genel olabilir veya dünyanın herhangi bir yerindeki bir sunucuda bir dokümanın yeri olabilir.

Bir internet veya web adresi URL olarak adlandırılır. Tipik olarak dört kısımdan oluşur:

- İletişim kuralı adı: İletişim kuralı, bilgisayarların bilgileri değiş tokuş etmesini sağlayan kurallar ve standartlar topluluğudur ve http kullanılmaktadır. ‘url: http// ’ ile başlamaktadır.
- Sitenin yeri : Siteler genelde www üzerindedir.
- Siteyi çalıştıran kuruluşun adı : Örneğin ‘www.itu.edu.tr’ adresindeki ‘itu’ adı gibi.
- Kuruluşun türünü tanıtan bir son ek (ticari kuruluşlar için .com, akademik kurumlar için edu, gibi)

5.1.5 Web Tarayıcısı

Bir web tarayıcısı, web sunucusuna bağlanan, orada gösterilen HTML kodlarından oluşan belgeyi alıp okuyan ve okuduklarını yorumlayıp kullanıcıya bunları düzenli ve bir sayfa düzeni biçiminde ekrana getiren bir programdır. Internet Explorer ve Netscape Navigator iki ayrı web tarayıcısıdır. Daha başka tarayıcı programlar da vardır, ancak şu anda internet dünyasında söz sahibi değiller. [23]

5.1.6 Web Sunucusu ve Internet Servis Sağlayıcılar

Web Sayfası, bir web sunucusundan alınan adrese, FTP programı kullanılarak sayfa bilgilerinin yüklenmesi ile yayınlanır. Aslında kişiler web sunucu olarak kendi makinalarını da kullanabilirler, ancak internette yayınlanan bir sayfanın sunucu bir işletim sistemine sahip bir bilgisayara kurulması ve her zaman ulaşılabilir olmak için 24 saat açık bırakılması gerekecektir. Ancak bu şekilde dünyanın her yerinden

kullanıcılara gece gündüz hizmet verebilsin. Bu nedenle, web sunucu hizmeti veren özel servislerden yararlanılır. Bu internet servis sağlayıcılara, ISP (Internet Service Provider) denir. Bir ISP, bir bilgisayar üzerinde farklı kuruluşlara ait ve herbirinin kendine ait bir domain adı ve IP adresi bulunan web siteleri bulundurabilir. Bir ISP, farklı müşterilere ait Web ve FTP siteleri oluşturmak için Internet Servis Yöneticisini (IIS) kullanır.

5.1.7 HTML Dosyaları, HTML Dili ve Kodlar

Web Sayfaları aslında birer ASCII (American National Standard Code For Information Interchange) metin dosyalarıdır. Bu dosyaların uzantıları .HTM veya .HTML'dir. Bu dosyalar metin dosyalarıdır, dolayısı ile bir metin editörü ile düzenlenebilirler veya baştan bir metin editörü ile oluşturulabilirler. Piyasada çok sayıda HTML editörü bulunmaktadır. HTML dosyaları, web sayfasında yer alacak olan metin ve bu metnin bir sayfa düzeninde yerleştirilmesini sağlayacak olan kodlardan oluşmuştur. [23]

Internet ve web sayfası kavramı ilk ortaya atıldığında, Amerikalı uzmanlar, kullanıcılara metin, resim, ses, film vs. gibi görsel ve işitsel elemanları bilgisayar ekranında sergileyebilmek için, ortak bir dil üzerinde karar kılınması gerektiğini farkettiler. Internetin maliyeti yüksek olduğu için, bu ortamı kullanıcısı bol bir hale sokmak gerekiyordu ki, sürüm artsın, yapılan işlerden kazanç sağlansın. Sonunda ortak bir karar ile HTML doğdu. Bu dil son derece esnek, basit ve geliştirilmeye açıktır. HTML dili, bir web sayfasını tasarlamak ve hazırlamak için kullanılan bir kodlama dilidir.

Bugün kullanılan HTML dilinin ilk hali Tim Berners-Lee tarafından, İsviçre'deki CERN laboratuvarlarında ortaya çıkmıştır. Bugüne kadar devamlı geliştirilen HTML dili daha önceleri sadece düz metinler ve birkaç resim formatı destekleyip görüntüleyebilirken bugünlerde yeni komutlarında eklenmesi ile oldukça gelişerek daha farklı ve geniş dosya çeşitlerini ve dosya formatlarını destekler hale gelmiştir. Böylece internet üzerinde oldukça çekici ve gelişmiş sayfalar hazırlanabilmesi mümkün olmuştur. [22]

HTML sayfası hazırlanırken kodların yazım kurallarına dikkat edilmelidir. Aksi takdirde web tarayıcısı kodları tanıyamaz. Özellikle her kodun < > parantezleri içine

yazılması gerekir. Bir kod, fonksiyonunun etkin olması istenen bölümün başına <KOD> şeklinde yazılır, ve etkinliğin bitmesi istenen noktada </KOD> yazılarak komut kapatılır.

5.1.8 JavaScript, VBScript, ASP Dili ve ActiveX Teknolojisi

JavaScript, Netscape firması tarafından C dilinden türetilmiştir. JavaScript bazen Java ile karıştırılmaktadır, bu yüzden bu konuya açıklık getirmek gerekmektedir. JavaScript, HTML dili gibi veya VBScript gibi, kodları düz yazı dosyası olarak kaydedilen ve işleyebilmesi için bir yorumcuya ihtiyacı olan bir dildir. Java ise Sun firması tarafından Pascal ve Delphi dillerinden esinlenerek yazılmıştır, ve bir Java programı işletim sisteminden bağımsız olarak kendi başına çalışabilir. [24]

Netscape, web tarayıcı programını ikinci sürümüne hazırlarken, HTML 'nin bir dil olarak yetersiz kaldığı yönleri gördü. Bir web sayfasına bir iş yaptırmak istediği zaman bunu HTML yaptıramıyordu. Ayrıca HTML yeteri kadar dinamik değildi. Mesela web ziyaretçisi bir sayfada sunulan, kentteki etkinlikler programını tarih sırasına göre almak istediğinde ancak bir CGI (Common Gateway Interface) programı yazması gerekiyordu. Ayrıca CGI, web ziyaretçinin tarayıcısı ile sitenin bulunduğu sunucunun bulunduğu noktadır. Burada çalıştırılacak programlar web sunucuda çalışıyor demektir. Her yeni ziyaretçi her yeni CGI programı web sunucuya ek yük yükler, bu da web sunucunun giderek yavaşlaması demektir. Siteleri dinamik hale getirmeyi amaçlayan ve web sayfasında çalışan bu tür programların web sunucuda değil de, ziyaretçinin bilgisayarında çalıştırılması fikrine göre hareket eden Netscape, LiveScript adıyla bir düzyazı dili geliştirdi. O sırada Sun firmasının geliştirdiği, PC veya Mac farkı gözetmeden, Windows, Unix, veya başka bir işletim sistemine tabi olmadan çalışacak bir dil üzerinde çalışıyor ve böyle bir dilin geliştirilmesi herkesi heyecanlandırıyordu. Netscape de 1995 Aralık'ta bu dili piyasaya JavaScript adıyla sürdü. Netscape 2.0'da JavaScript uyumlu olarak çıktı. Tabi Microsoft da, Internet Explorer adlı web tarayıcısına bu imkanı kazandırmak istedi, ancak 1 yıla yakın bir süre Netscape firması JavaScript'i paylaşmaya yanaşmadı. Microsoft, Netscape'in JavaScript yorumlama kodunu anlamadığı halde Internet Explorer'i JavaScript'i anlar hale getirdi, ve ortaya çıkan dile de Jscript adı verildi. Aslında JavaScript ile Jscript tümüyle aynı idi, ve her ikisi de hem JavaScript hem de Jscript anlıyorlardı. Bu durum fazla sürmeden her iki firmada kendi Script

dillerine diğerk programın tanımadığı özellikler kazandırdılar. Bu arada Microsoft Internet Explorer'a Jscript'te de JavaScript'te de olmayan becerilerle donattığı VBScript'i tanıma özelliği kazandırdı. Sonuçta, ortaklaşa bir dile gitmenin faydalı olacağına karar veren Microsoft ve Netscape firmaları Avrupa Birliği Standart kurumuna bir öneride bulundular ve JavaScript ve Jscript dillerini birleştirecek ortak 'ECMAScript' isimli bir standart yayınlandı. [24]

ASP (Active Server Pages) : Web tarayıcıdan bağımsız dinamik içerikler oluşturmak için sunucu tarafında scriptler ve bileşenler kullanılmasına olanak verir. Microsoft'un ürettiği, sunucunun aktif sayfalar sunma teknolojisidir. ASP ile JavaScript'in farkını anlayabilmek için bir uygulama göz önüne alınabilir. Örneğin, URL hanesine <http://www.damlauca.com.tr/index.asp> gibi bir adres girildiğinde, web sunucu bu asp sayfasını alır, doğrudan ziyaretçiye göndermez, önce içindeki kodları icra eder, sonra bu kodları ayıklar ve geriye saf ve temiz HTML kalır. Bu sayfa ziyaretçinin tarayıcısına gönderilir. Burada önemli olan 3 nokta vardır.1. ASP sayfasının içinde kod vardır. 2. ASP sayfasının kodları web sunucu tarafından icra edilir. 3. ASP sayfası, ziyaretçinin tarayıcısına sade HTML olarak gelir. Burada ASP ile JavaScript'in farkı görülmektedir. JavaScript, HTML içine konuluyor, ama sunucuda değil tarayıcı da çalışıyor. Bu aşamada CGI/Perl programı ile ASP'nin farkı da görülmektedir. HTML sayfasını Perl ile yazılan program üretiyor, oysa ASP'de ziyaretçiye gidecek HTML'yi sunucu ve tasarımcı birlikte üretir. Bu bakımdan, JavaScript istemci tarafında çalışırken, ASP, CGI programları gibi sunucu tarafında çalışır. Bunun sonucu ise, ASP ile yazılacak sayfaların, ziyaretçinin bilgisayarında kurulu tarayıcı programı ne olursa olsun mutlaka görüntülenmesidir. ASP sayfalarına koyulan VBScript kodları sunucuda icra edileceği için, ziyaretçinin tarayıcısının VBScript'i tanımayan Netscape olması ziyaretçiyi etkilemeyecektir. [25]

ASP kodları HTML kodları ile beraber yazılır. ASP kodlarının içinde JavaScript, VBScript, Jscript olabilir. [26]

ASP dosyaları da HTML dosyaları gibi ya bir editör ile ya da MS Front Page veya MS Visual InterDev gibi bir yazılımı kullanarak geliştirilir. ASP dosyalarını çalıştırabilmek için, IIS (Internet Information Services) programı gereklidir.

ASP teknolojisinin en önemli özelliklerinden birisi de veritabanına erişim sağlayan web sayfaları sunmasıdır.

ActiveX, Microsoft'un 1996 yılında sunduğu internet araçları kümesi ile karşımıza gelen bir teknolojidir. ActiveX, farklı firmalar tarafından üretilen yazılım komponentlerini aynı işlemde, aynı makinada veya birçok makinada dağıtılmış halde kullanımına izin verir. [26]

5.1.9 IIS Yazılımı ve Yönetimi

IIS (Internet Information Services): TCP/IP protokolünü kullanarak internet veya intranet üzerindeki web sitelerine ev sahipliği yapmayı sağlar. Microsoft işletim sistemlerinde, WWW, FTP ile dosya aktarımı ve posta kullanımını sağlar. Standart bilgi protokolleri ile hem internet sunucusu uygulama programlama arayüzü (ISAPI) hem de CGI kullanımını destekler. IIS; internet, intranet ve extranet siteleri için bir sunucu çözümü sağlar. IIS; Microsoft Visual Basic Programlama dilini, VBScript, Microsoft Jscript geliştirme araçlarını ve Java bileşenlerini tamamen destekler. IIS, ayrıca web tabanlı programlar için CGI uygulamalarına ve ISAPI uzantılarına tamamen destek verir. [26]

5. 2 KBS'NİN WEB SUNUCU SİSTEME KURULMASI

5.2.1 KBS Verisi

Kent Bilgi Sistemi olmak üzere, bir CBS yazılımı kullanarak hazırlanan verinin, bir web sunucu üzerinde bulunması gerekir. Kent Bilgi Sistemini oluşturan tüm verinin aynı klasör içinde bulunması gerekmez. Veri çok ise, buna bağlı olarak dağıtılmış olabilir. Ancak, kent bilgi sistemini oluşturan veriler ile bunları web ortamında sunmak için hazırlanan dosyaların, yani projeyi oluşturan tüm verinin bir klasörde bulunduğu varsayılarak, metnin devamında 'veri klasörü' olarak bahsedilmiştir.

Veri klasöründe, pek çok türde dosya bulunacaktır. Bu klasörde, bilgi sisteminin coğrafi içeriğini oluşturan çizim dosyaları, nitelik bilgilerini saklayan veritabanı, web sayfasını oluşturan HTML ve görsel içerikli .JPEG türü dosyalar ile web sayfasını veritabanına bağlayan ASP dosyaları bulunur.

Web üzerinden sunulacak verinin coğrafi bileşene sahip olması önemli bir noktadır. Verinin geometrik olarak ifade edilmesi gerekmektedir ve bunu sağlamak yolunda iki çözüm getirilmiştir:

Birinci yol olarak, sistem CBS yazılımı kullanılarak oluşturulan CAD dosyalarını veri klasöründen alır. Bu çizimler katman yapısında saklanmışlardır, ve bir program aracılığı ile çağırılarak web ortamına aktarılırlar.

İkinci yol olarak ise şudur: Çizim yapılırken oluşturulan veritabanında, ilgili kent ögesine ait tabloda ayrılan bir sütuna her elemanın geometrik bileşeni otomatik olarak işlendiği için, sistem bu sütundaki verileri kullanarak harita üretebilecektir. Bu nedenle, sunucu herhangi bir CAD dosyasına ihtiyaç duymadan, veri klasöründeki veritabanı dosyasından yararlanarak harita oluşturmaktadır.

Bu tez kapsamında yapılan uygulamada, ikinci yol seçilerek, çizim dosyaları değil, veritabanındaki geometri kayıtları kullanılarak coğrafi nesneler oluşturulmuştur.

Ayrıca, KBS projesini internetten sunmak isteyen proje sahiplerinin, çok farklı CBS veri kaynaklarına sahip olabileceği unutulmamalıdır. Seçilen yazılım, farklı CBS veri kaynaklarından yararlanarak harita oluşturabilecek yetenekte olmalıdır.

5.2.2 GeoMedia Web Map Yazılımı, ACGM Kontrolü

GeoMedia® Web Map programı, Intergraph Şirketi'nin internet veya intranet üzerinden harita ve coğrafi bilgi sistemleri verileri yayınlamak üzere ürettiği sunucu bir programdır. Windows NT Server™ ve Windows® 2000 Server işletim sistemleri ile uyumlu olarak hazırlanmıştır. Bu yazılım bir CBS programı değil, CBS verisini internet üzerinden sunma programıdır.

5.2.2.1 Web-Sunucu Donanımı

Intel chip tabanlı sunucu sistemleri ile uyumlu çalışmak üzere dizayn edilmiş bu yazılım, veritabanı donanımı konusunda ise Intel tabanlı sistemlerle sınırlandırılmamıştır.

Web sayfası üzerinden veri indirilirken, süre çok önemlidir. En iyi performansı almak için farklı donanım gereksinimleri duyulabilir. Aynı anda 12 kullanıcıdan fazla ziyaretçinin bağlanmasının beklendiği bir site için, tek işlemci yeterli olmayacaktır.

5.2.2.2 Web-Sunucu Yazılımları

Web sunucu makinada, sunucu bir işletim sistemi ile web sunucu platformu bulunması gerekir. Uygulamada, Windows 2000 Server İşletim Sistemi ile web sunucu platformu olarak IIS 4.0 kullanılmıştır.

Ayrıca programı tamamlamak üzere bazı yazılımlar kullanılmıştır. Web sayfası tasarım araçlarından Microsoft Visual InterDev'den yararlanılmıştır. Hazırlanan ASP dosyaları içerisinde script dili olarak Visual Basic™ Script kullanılmıştır. Ayrıca Jscript, JavaScript veya Perl gibi uygulamalar da mümkündür.

İstemci tarafta da olması gereken minimum şartlar şunlardır: İşletim sistemi olarak Microsoft Windows® 98, Windows NT® 4.0 ve üzeri, veya Windows® 2000, web tarayıcı yazılımları olan IE ve NN, ActiveCGM plug-in'i olarak, Netscape Navigator için Micrografx® Inline plug-in ya da Internet Explorer için Micrografx® ActiveX bileşeni yüklenmelidir.

5.2.2.3 ActiveCGM Kontrolü

GMWM programı, ActiveCGM™ formatında harita üretmektedir. CGM (Computer Graphics Metafile), ISO (International Standards Organization) tarafından, grafik gösterim formatında bir standart olarak tanımlanmıştır. Bu standart, tanımında uygulamalar sırasında eklenen grafik ayrıntıları veya CGM standardının yürütülüşüne özel olan metadata'yı tanıyabilme seçeneğini de içermektedir. Micrografx®'in InterCap birimi, kendi geliştirdiği ve yürüttüğü uygulamalar için bu standardı sağlayan CGM metaverisini tanımlamış, ve ActiveCGM™ olarak isimlendirmiştir. GMWM, ActiveCGM (ACGM) formatlı haritaları göstermek için Micrografx web tarayıcı kontrollerini kullanmaktadır.

Bir ActiveCGM dosyasını görüntülemek ve etkileşimli olarak çalıştırmak için web sayfası içerisine bir kontrol yerleştirmek gerekmektedir. Hem harita üretecek sunucu makinada, hem de istemci makinada bu kontroller yüklü olmalıdır. Web tarayıcısı olarak Microsoft Internet Explorer kullanan istemci makinaya ActiveX kontrolü yüklenmelidir. Web tarayıcısı olarak Netscape'i tercih eden istemciler için de kullanılacak bir plug-in mevcuttur. Her ikisinin de harita üreten web sitesi üzerinden istemcilere yüklenmek üzere sunulmasına herhangi bir para talep edilmeden imkan verilmiştir.

ACGM dosyası web tarayıcısında görüntülenirken, kullanıcı haritayı yaklařtırma, harita üzerinde iřaretleme yapma ve yazı yazma, yazıcıdan çıktı alma gibi bazı fonksiyonlardan yararlanabilir. Bu fonksiyonlar haritanın ACGM formatında üretilmesi ile saęlanmaktadır. Web sayfasında fareyi, üretilen haritaların yerleřtirildięi alana taşıyarak saę tuřa basıldıęında, kullanılabilir fonksiyonların olduęu bir menü ekrana getirilir.

ActiveCGM gösterimi, zeki grafik üretme yeteneęine sahiptir. Raster veri ve vektör veri gösterir, harita elemanlarına fonksiyon kazandırır, etkileřimli harita oluřturur ve harita elemanlarına etki kazandırmak üzere programlanabilir. Burada bahsedilen programlama, multimedya bir dosya görüntülemek, bařka bir URL adresine baęlanmak, veya bir harita elemanına tıklayarak hizmete sunulan veritabanına baęlı bir SQL sorgu yapmaktır. Bu sorguların sonuçları, haritanın tekrar üretilmesi ve yeni gösterimde sorgu sonuçlarına iřaret eden bir haritanın görüntülenmesi ile sunulmuř olur.

5.2.3 Sistemin Kurulumu

Sistemde ařaęıda verilen kuruluř ařamaları gerçekteřtirilir:

1. Internet Information Server yüklenir ve sunucu hizmeti ile ilgili ayarlar yapılır.
2. GeoMedia Web Map programı yüklenir.
3. Kullanılacak ve internet üzerinden yayınlanacak KBS verisi, verilmek istenen hizmete göre hazırlanır.
4. Web dizaynı yapılır, sitede hizmet verecek web sayfaları hazırlanır.

5.2.3.1 IIS Yüklenmesi ve Ayarlar

IIS, web sitesi yayınının yönetimi amacıyla kullanılan bir programdır. IIS programı ile, sunucu olarak kullanılacak makineye internet ortamında sunulacak olan projenin klasörleri tanıtılmıřtır.

IIS programı altına internetten hizmet vermek üzere proje klasörü olarak veri klasörü, ve cache klasör olarak boş bir klasör tanıtılır. Proje klasöründen veriler kullanılacaktır, cache klasörü ise geçici dosya depolamak üzere kullanılacaktır.

Klasörler belirtildikten sonra IIS'de sunucu özellikleri ile ilgili bazı ayarların yapılması gerekir.

5.2.3.2 GMWM Yüklenmesi ve Kurulması

GeoMedia Web Map programı CD'den yükleme talimatları izlenerek kurulduktan sonra, program yetkilisi araçlarından sunucu ayarları yapılır. Bunlar cache klasöründe verinin tutulma süresi, harita sunucunun time out süresi, aynı anda çalışabilecek maksimum harita sunucu sayısı ve üzerinden yayın yapılacak root URL adresinin ne olacağı gibi ayarlardır.

5.2.3.3 Verinin Hazırlanması

Coğrafi Bilgi Sistemleri ile bir bölgeye ait coğrafi veriler sayısallaştırılır, CBS ortamında farklı veri kümeleri farklı katmanlara atılır, oluşturulan vektörel harita elemanları ile bu elemanlara ait öznitelik bilgileri bağlantılandırılır, bu sırada program tarafından veritabanına kaydedilir. CBS verisi, bir çok CBS programı kullanılarak farklı formatlarda hazırlanabilir.

Projeyi Oluşturan Veri Kaynaklarına Bağlantı: Web sayfasında gösterilen grafik elemanların veri klasöründe çizim dosyası veya veritabanında kayıt olarak bulunabileceği belirtilmişti. Bu depolanan veriye bağlantı kurulması gerektiği gibi, her bağlantının da, verileri görüntülenebilecek bir formata çevirmesi gerekir. Kullanılan yazılım, aşağıdaki veri formatlarına bağlantı kurmak için uyumludur:

- Access
- ARC/INFO®
- ArcView
- CAD (MicroStation veya AutoCad)
- FRAMME
- GeoMedia SmartStore
- MapInfo®
- MGE
- MGE Data Manager (MGDM)

- MGE Segment Manager (MGSM)
- ODBC Tabular
- Oracle® Object Model
- Oracle® Relational Model

MGE dosyalarına, ODBC Tabular verisine ve veritabanına link bulunduğunda Microstation çizim dosyalarına ODBC bağlantısı yapmak gerekmektedir. Bunun dışındaki her veri kaynağına doğrudan bağlantı yapılabilir.

Program, farklı kaynaklardan da olsa, bu veri sunuculardan gelecek verileri tek bir harita üretmek üzere birleştirme olanağı sağlamaktadır.

Bu tez kapsamında gerçekleştirilen uygulamada, sunucu makinesinin KBS'ye ait geometriyi, sadece veritabanından oluşturma yolunu seçtiği belirtilmişti. Bu durumda, bir makinede hazırlanan kent bilgi sistemi projesinden yalnızca veritabanını alarak web sunucu makinesindeki proje klasörüne atmak yeterli olmuştur. Proje klasörüne ayrıca haritaların koordinat sistemi bilgisini almak üzere, GMWM'te oluşturulan bir .csf (Coordinate System File) dosyası da eklenir.

Proje internet ortamına aktarılırken, bu veri ile ne tür analiz ve sorgulamaların yapılacağı konusunda, bilgiler çok dikkatli bir şekilde gözden geçirilmelidir. Şu tür sorulara cevap aranarak site planlanmalıdır:

- KBS içinde hangi veriler mevcuttur?
- Hangi veriler ulaşılabilir olmalıdır?
- Veri ne şekilde sunulmalıdır?
- Performans ve güvenlik göz önüne alındığında kullanılacak veri kümeleri nelerdir?
- Veriler harita sunum hizmeti veren makina üzerinde mi olmalı yoksa uzak bir makineye bağlanıp, asıl verinin bir kopyası mı kullanılmalı?

5.2.3.4 Web Sayfası Tasarımı

Projenin önemli bir kısmı ise, sitenin ziyarete uygun, kullanışlı ve görsel anlamda tatmin edici tasarlanmasıdır. Arka planda ne kadar güçlü bir sistem kurulmuş olursa olsun, istemcinin kullanacağı arayüzün etkili, uyumlu ve kullanımı kolay olması gerekmektedir. Örneğin web sayfası üzerinden, kullanıcının haritaları görmesi için gereken ACGM kontrolü kolayca istemci makineye yüklenmelidir.

Web sayfaları haritaların sunulacağı ortamdır. Web tasarımcısı, genel görünümü ve bağlantılarla açılacak sayfaları dinamik bir kullanıma uygun olarak tasarlamalıdır.

Web sayfası tasarlanırken aşağıdakilere dikkat edilir:

- Amaç web sitesinde neyin sunulmasıdır?
- Web sayfalarında görüntülenecek olanlar nelerdir?
- İstemciler ne tür bilgilere ulaşmak isteyebilirler ve bu bilgiler nasıl sağlanmalıdır?
- Web sayfasında istemci nasıl yönlendirilmek isteniyor? Buna bağlı olarak ne tür bir arayüz kullanmak uygun olur?
- KBS elemanlarından hangileri kullanılacak ve nasıl bir semboloji ile gösterileceklerdir?
- Veriyi sunarken ne şekilde fonksiyonlara sahip olmaları isteniyor?

Web sitesi için, görünüme yönelik genel hatlar tasarlandıktan sonra, haritalarda ne tür fonksiyonların olacağına karar verilmelidir. Projeyi oluşturan web dosyaları içinde HTML, ASP, ve script programlama dilleri kullanılarak, haritalara fonksiyonellik kazandıracak programlar yazılır.

Programın mantığı incelendiğinde görülmektedir ki, statik haritalar kullanılmamakta, istemcinin seçtiği haritalar sunucuda o anda üretilmektedir. ASP dosyası, internet üzerinden bağlanan kullanıcının talebi üzerine ve taleplerine bağlantılı olarak çalışır, istenilen haritanın üretilmesini ve bu haritanın istekte bulunan kullanıcının web tarayıcısına gönderilmesini sağlar. Harita ACGM ile üretilcektir. Sistem, bu

özelliklere sahip olmayan kullanıcılara, .jpg formatında hazırlanmış sonuç haritaları göndermek üzere tasarlanabilir.

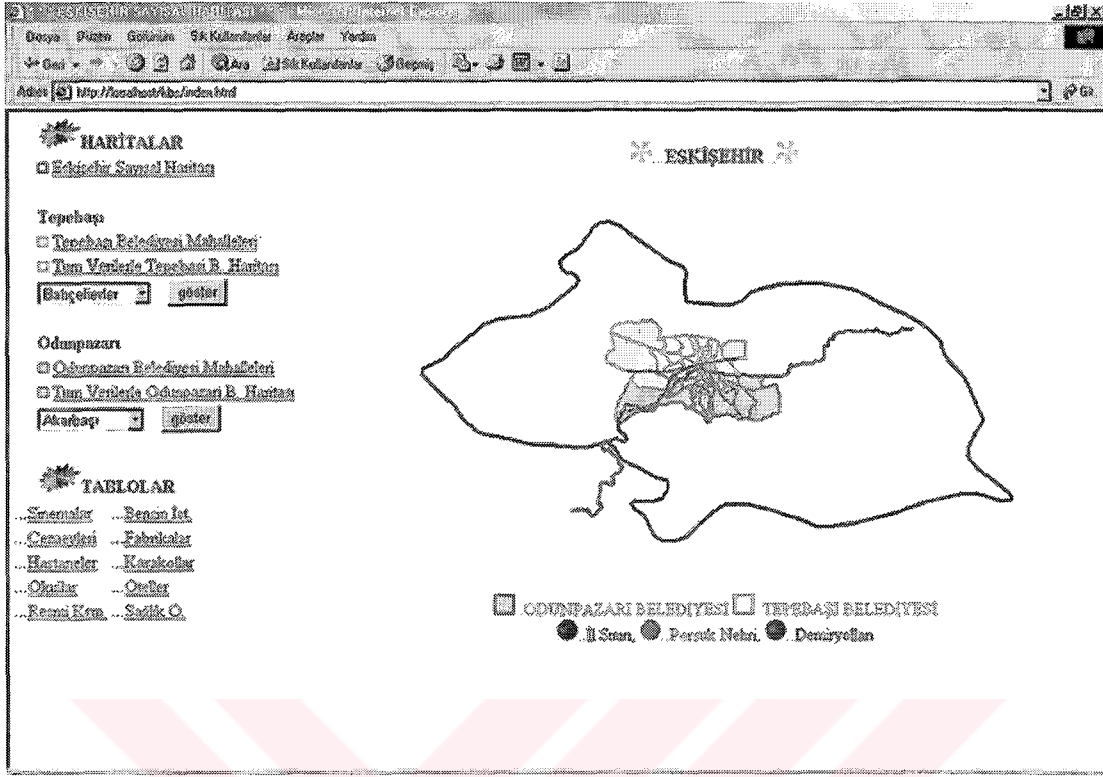
Harita fonksiyonları ve ürün haritanın sembolojisi ASP kodlama dili ile sisteme aktarılacaktır. GeoMedia Web Map programının aynı işlemi yerine getirmek için kullanılabilen bir arayüzü de mevcuttur. MDF (Map Definition File) adı verilen bu arayüz sayesinde her veri grubunun gösterim şeklinin tanımlandığı bir dosya hazırlanır ve bu dosya proje klasöründe yer alır. Bu uygulamada, ASP kodlama dili ile semboloji ve fonksiyonlar hazırlanmıştır.

Web sayfa tasarımcısının bu imkan ve araçların dışında, HTML, Dynamic HTML, VBScript, JScript, JavaScript ve diğer ActiveX kontrolleri kullanılarak görsel ve fonksiyonel anlamda sayfalarını zenginleştirmesi mümkündür.

5.2.4 Sistemin Çalışması

Kullanıcılar, internete bağlandıktan sonra, harita sunmak üzere hazırlanmış siteye, standart web tarayıcıları olan IE (Microsoft Internet Explorer) veya (NN) Netscape Navigator'ı kullanarak girecek ve sunulan KBS verisini görüntüleyebileceklerdir. İstemcinin IE veya NN arayüzünde görüntülediği haritalar ACGM ile üretilirken, KBS veritabanındaki bilgiler kullanılmaktadır.

Şekil 5.1'de kent bilgi sisteminden bir görüntü görülmektedir. Burada tüm kent genel harita olarak gösterilmektedir. Harita ACGM formatındadır.

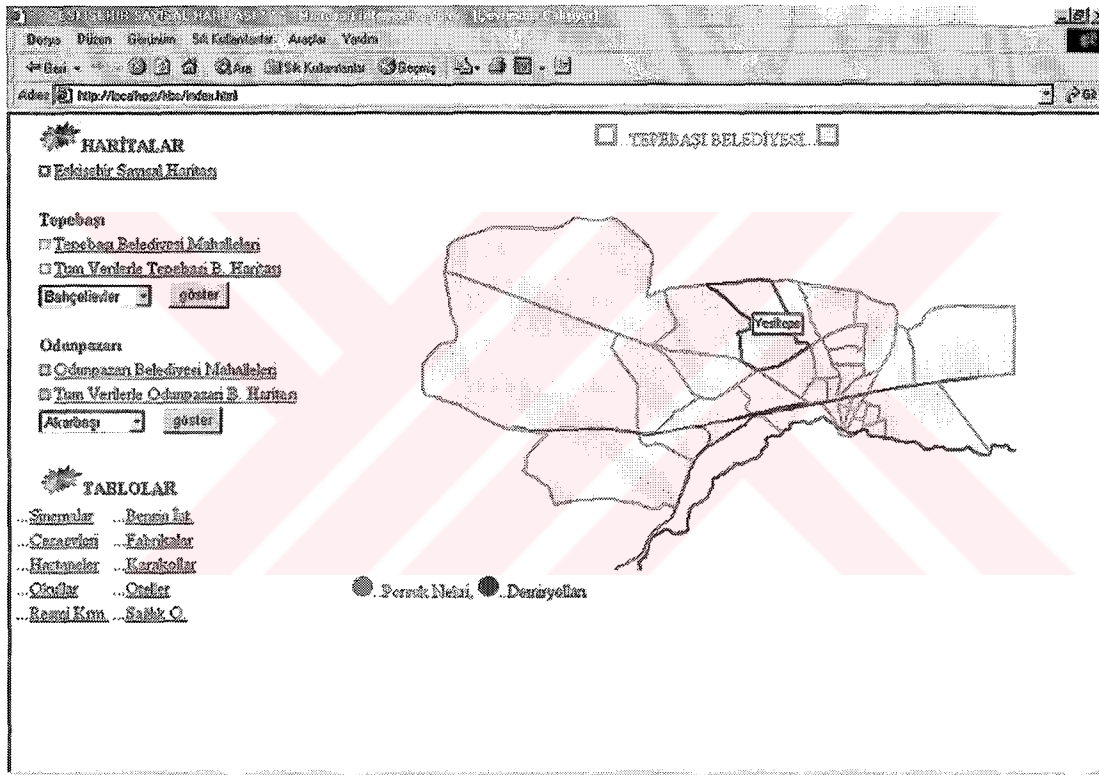


Şekil 5.1 Eskişehir Kent Bilgi Sisteminde genel Eskişehir görüntüsü

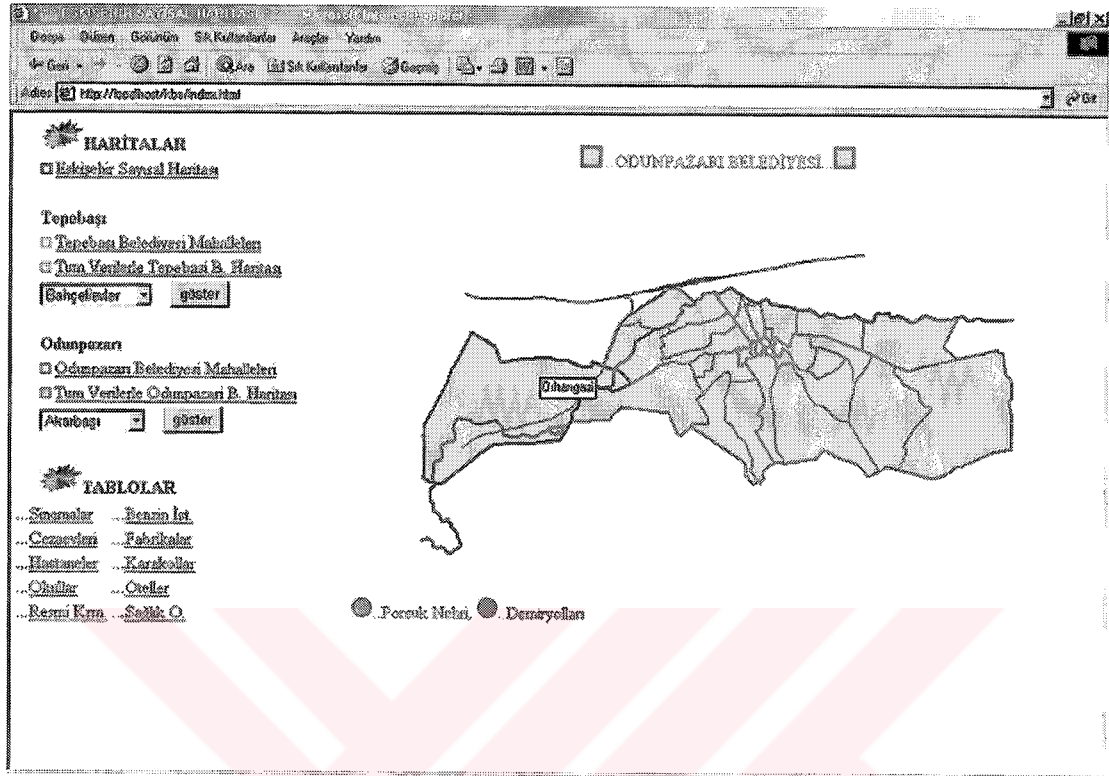
Her geometrik elemanın, veritabanında geometrik özellikleri ve öznitelik bilgileri bir arada tutulmaktadır. ASP kodları ile komut verilerek GeoMedia Web Map'in harita sunu prosesi başlatılır. Harita sunu işlemi boyunca, veri ile bağlantı kurulur, elemanların kayıtları okunur, gösterimleri ve kazandırılacak fonksiyonlarına göre işlemleri yapılarak, ürün harita ortaya çıkarılır. ASP dosyalarında, web sayfasının düzeni HTML kodları ile oluşturulmuş ve harita üretimi de script komutları ile sağlanmaktadır. Şekil 5.1'deki harita üzerinde fare ile dolaşırken vektörel çizim olduğundan nesneler tanınmakta ve ayrı ayrı algılanmaktadır.

ActiveCGM veritabanındaki öznitelik verilerine linkli geometrik elemanlardan oluşan, zeki haritalar üretilmesini sağlar. Mouse ile nokta, çizgi, poligon, yazı, veya sembollerden oluşan grafik elemanların üzerlerine gelince, bu elemanlara ait veritabanında kayıtlı bazı özelliklerin gösterilmesi, tıklanınca ilgili bir sayfaya bağlanması gibi fonksiyonlar akıllı harita özellikleridir.

Şekil 5.2’de Tepebaşı Belediyesi’ne, şekil 5.3’te ise Odunpazarı Belediyesi’ne ait haritalar görülmektedir. Bu haritalarda da dikkat edilirse farenin üzerinde olduğu nesne farklı bir renkle gösterilmiştir, ve ayrıca sistem veritabanından mahalle adı sütununa bu nesne için karşılık gelen değeri bulmakta ve görüntülemektedir. Bu etiketlemenin avantajları çoktur. Hem harita üzerinde farenin geldiği yerin adı verilmekte, hem de sınırlarını gösterilmektedir.



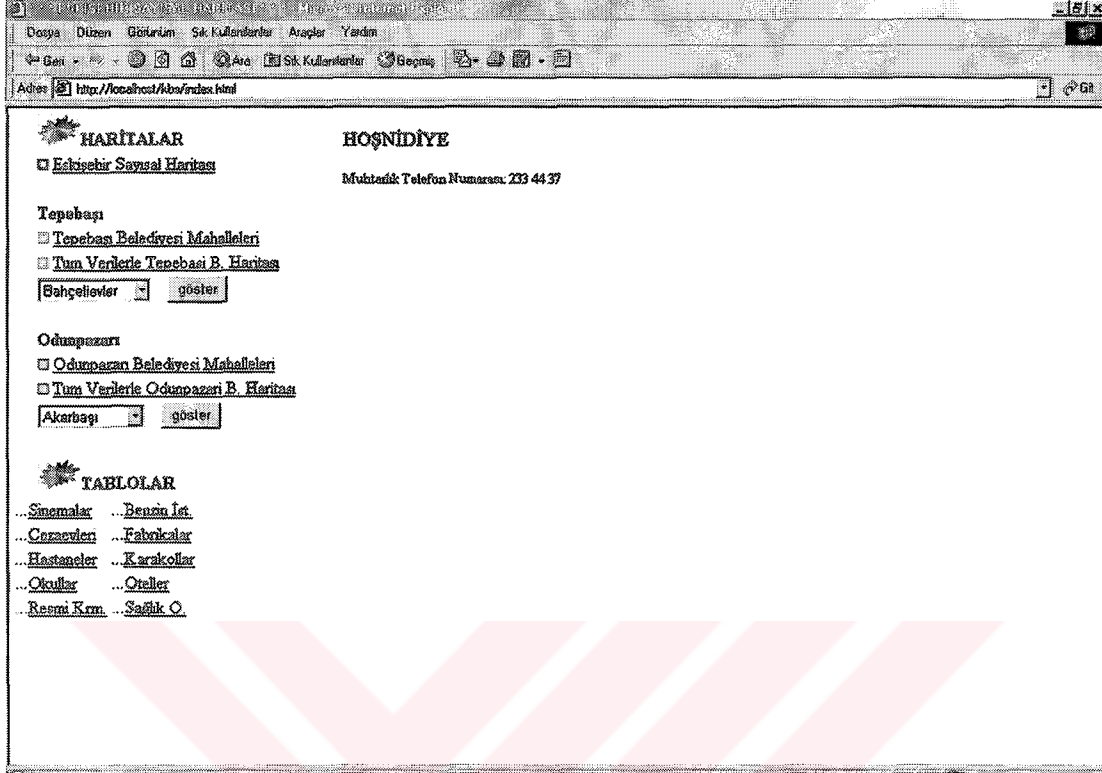
Şekil 5.2 Tepebaşı Belediyesi Mahalleleri



Şekil 5.3 Odunpazarı Belediyesi Mahalleleri

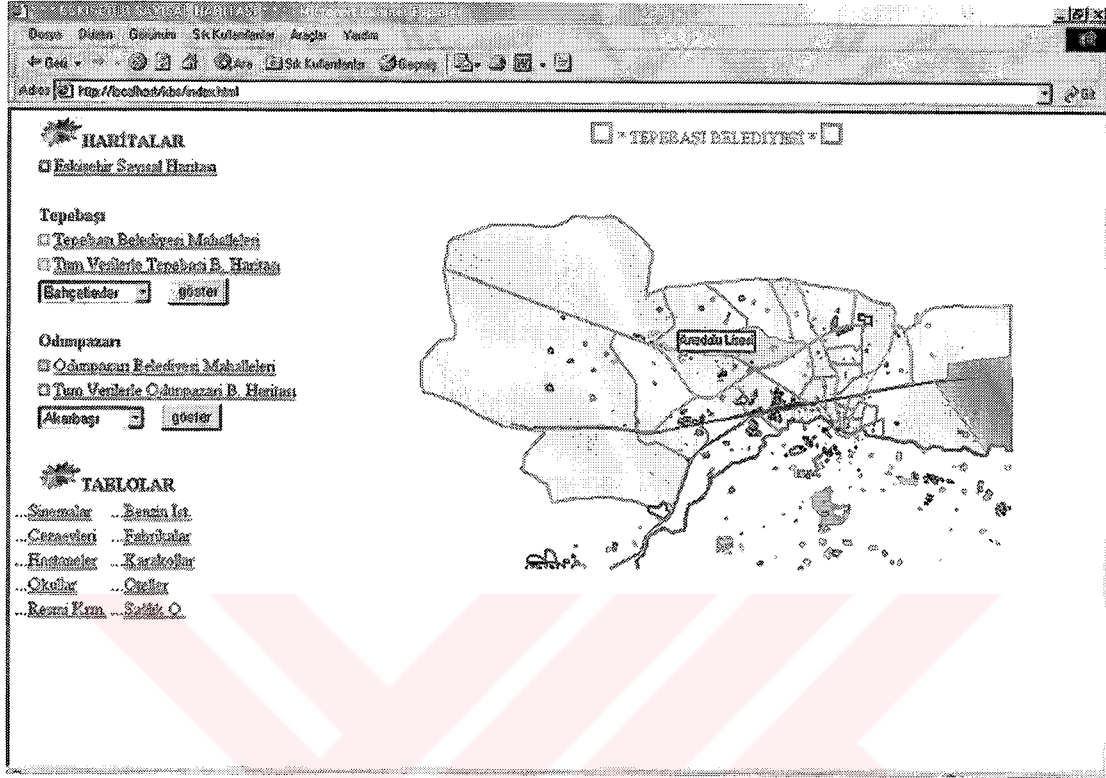
Harita nesnelerinden birinin üzerine gelindiğinde onunla ilgili bir sayfaya bağlanması da akıllı harita özelliklerinden biridir. Şekil 5.4'te Hoşnudiye Mahallesi üzerine gelindiğinde bu nesneye tıklanınca, o mahallenin muhtarlık telefon numarasının verildiği bir sayfa gelmektedir.

Kent Bilgi Sistemleri oluşturulurken, her muhtarlığın kendi mahallesi ile ilgili sunmak istediği bilgilerden oluşan veya mahalle ile ilgili yeni haberler içeren sayfalarına bağlanılmak üzere web sayfaları hazırlanabilir.



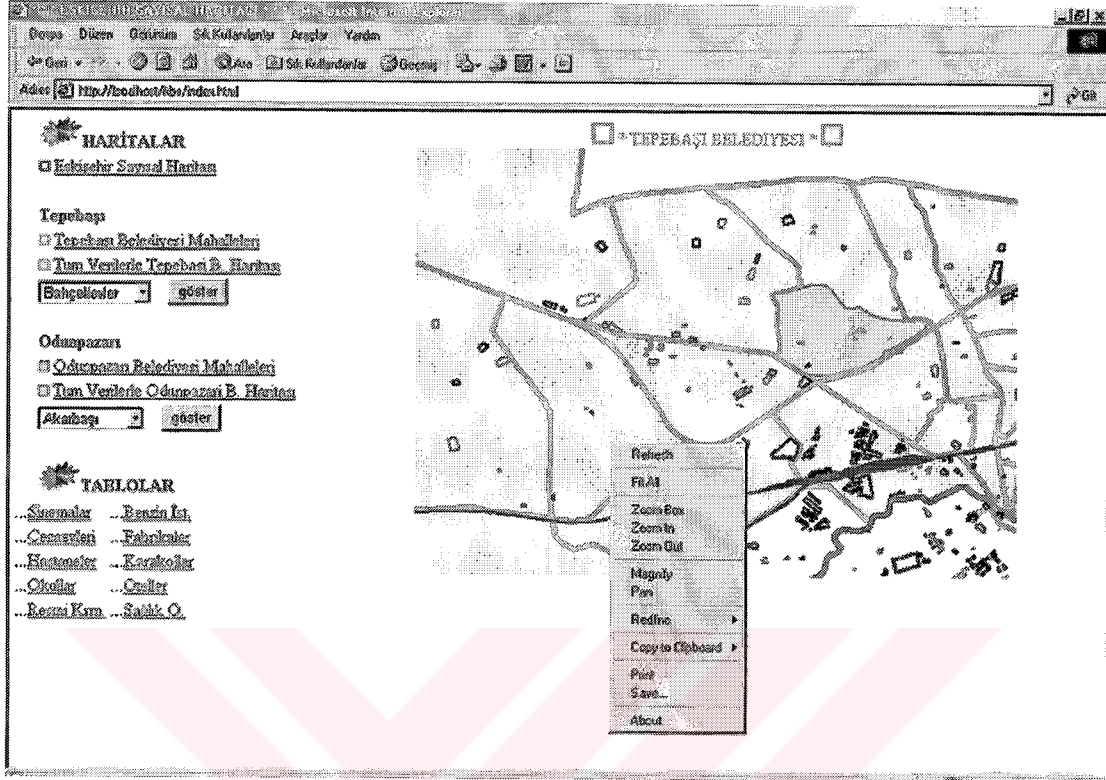
Şekil 5.4 Hoşnudiye Mahallesi'ne tıklayarak ulaşılan ilgili bir web sayfası

Şekil 5.5'te, Tepebaşı Belediyesi, tüm kent ögeleri ile görüntülenmektedir. Bu coğrafi elemanların üzerlerine gelinince isimleri gösterilecek şekilde etkileşimli hazırlanmışlardır.



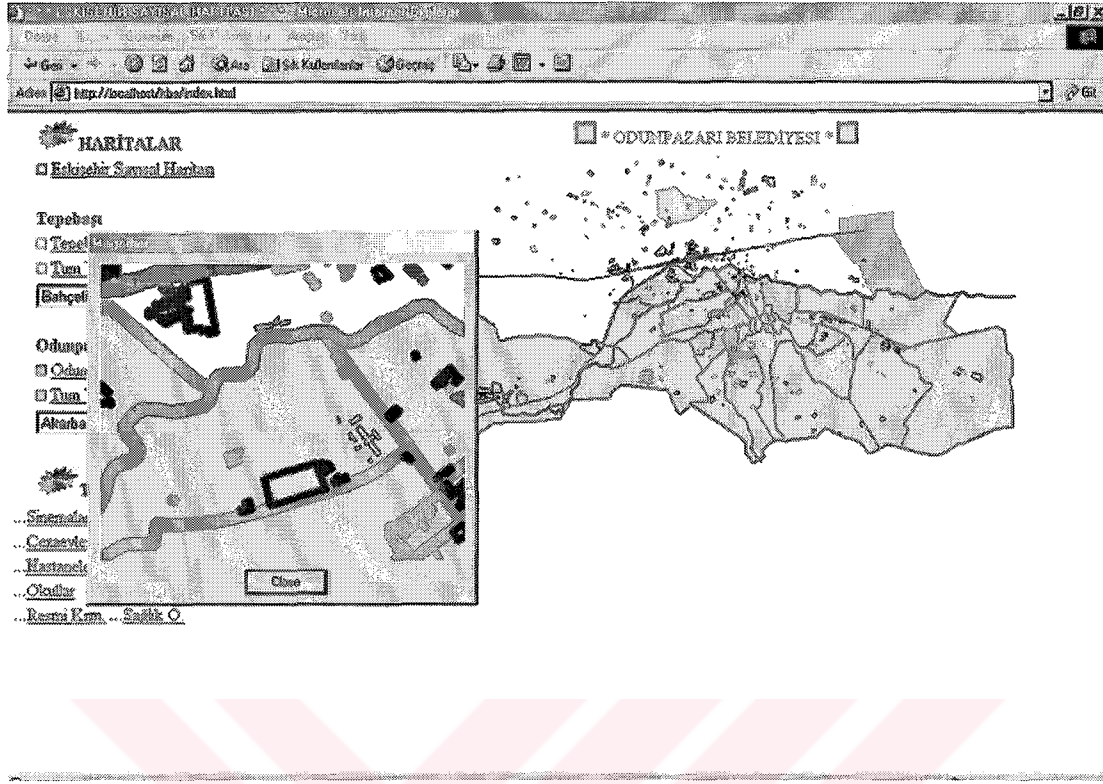
Şekil 5.5 Tepebaşı Belediyesi mahalleleri ve kent öğeleri

Bu şekilde bir harita ekrana getirildikten sonra ACGM kontrollerinden yararlanılabilir. Örneğin burada kent öğeleri ni daha büyüterek göstermek istenebilir. Şekil 5.6'da görüntünün, ACGM menüsünden yararlanarak yaklaştırılmış hali ve harita üzerinde sağ tuşa tıklanarak ekrana getirilen ACGM menüsü görülüyor.



Şekil 5.6 ACGM kontrolleri ve bunları kullanarak yaklaşıtırlan harita

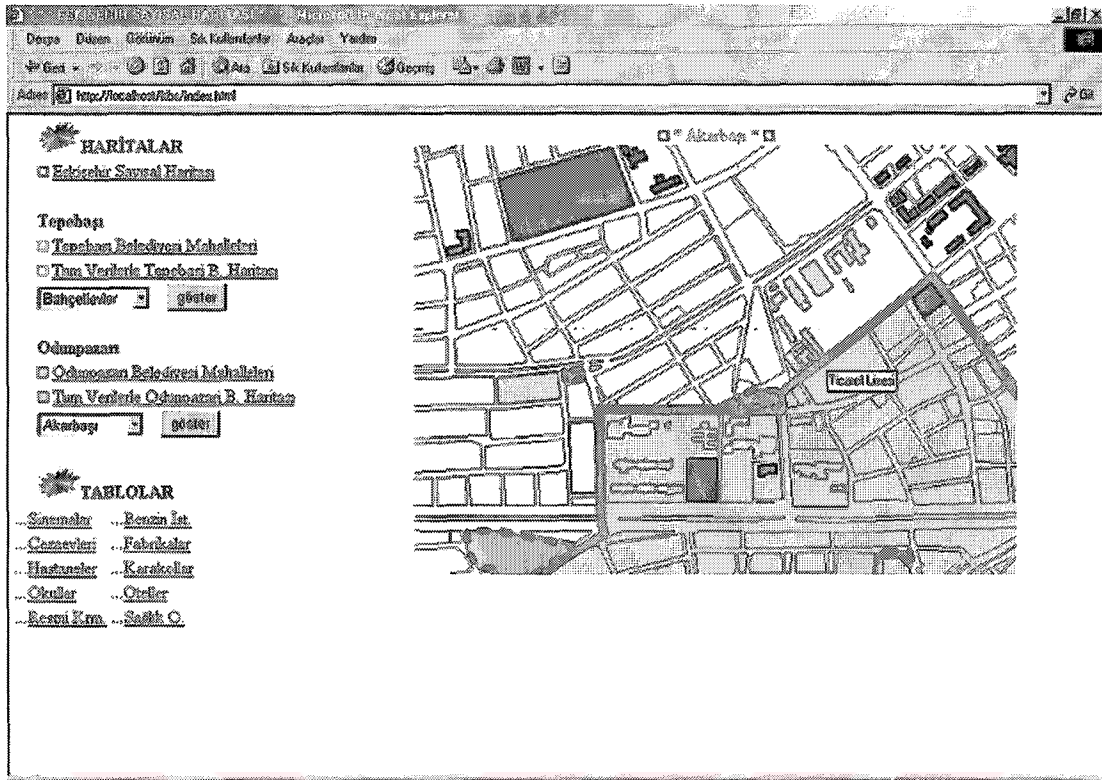
ACGM kontrollerinden önemli fonksiyonlarından birisi de mevcut haritayı izlerken küçük bir pencereden de yaklaşıtırlmış haritayı görüntülemeyi sağlamaktadır. Şekil 5.7’de bu şekilde çalışılırken alınan bir görüntü bulunmaktadır.



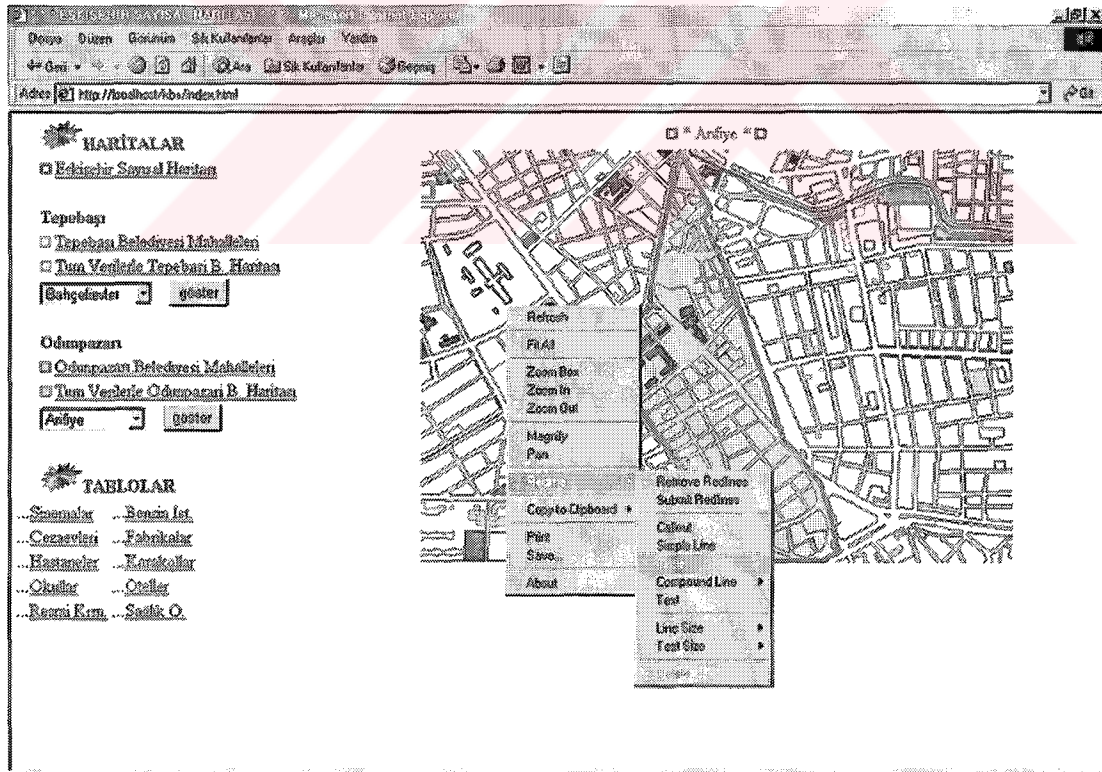
Şekil 5.7 Ekrandaki harita ve magnify penceresinin bir arada kullanılması

Sol framede görülen drop-down menü yardımı ile, mahalle boyutunda haritalara da ulaşılabilir. Bu haritalarda parsel boyutunda veri rahat görülebileceğinden, haritaya eklenmiştir. Şekil 5.8’de Akarbaşı Mahallesi görüntülenmiştir. Bu haritalarda da yine elemanların üzerlerine gelindiğinde isimleri gösterilmektedir.

Bu haritaların üzerinde yine ACGM kontrolleri kullanılarak, işaretleme yapılabilir, hatta üzerindeki işaretleme, çizim ve yazılar ile çıktısı alınabilmektedir. Şekil 5.9’da ACGM menüsünden yazı ve çizgi ayarları yapılmaktadır.

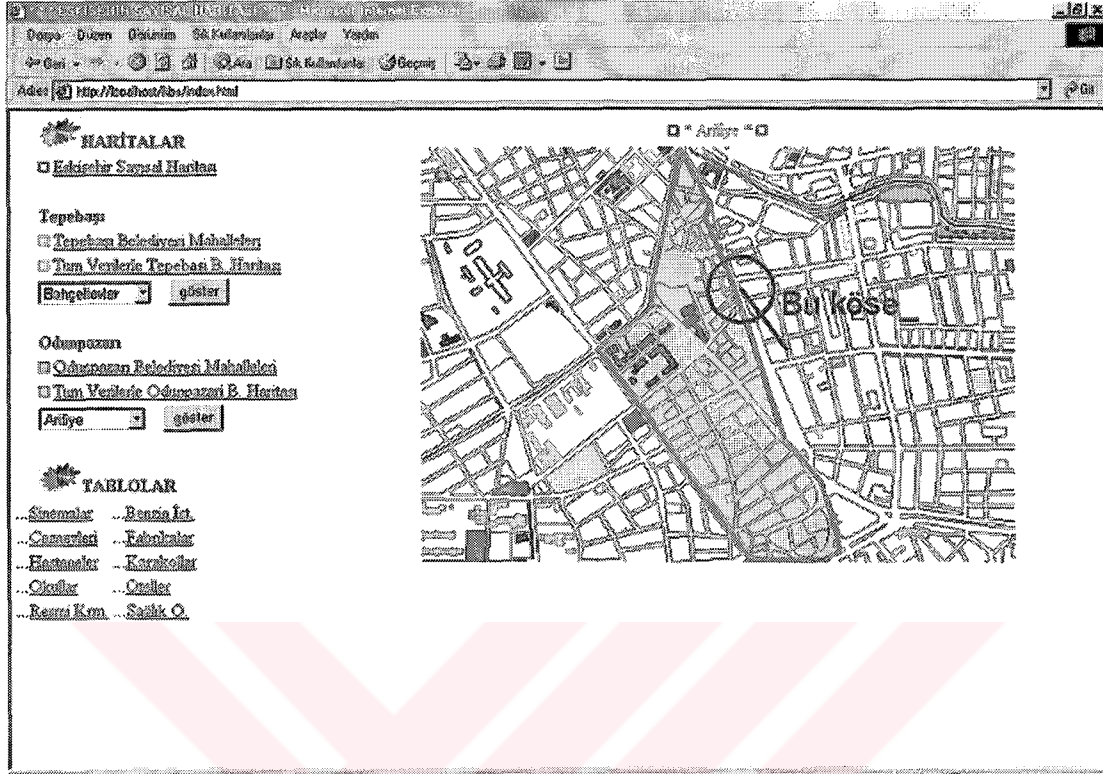


Şekil 5.8 Akarbaşı Mahallesi



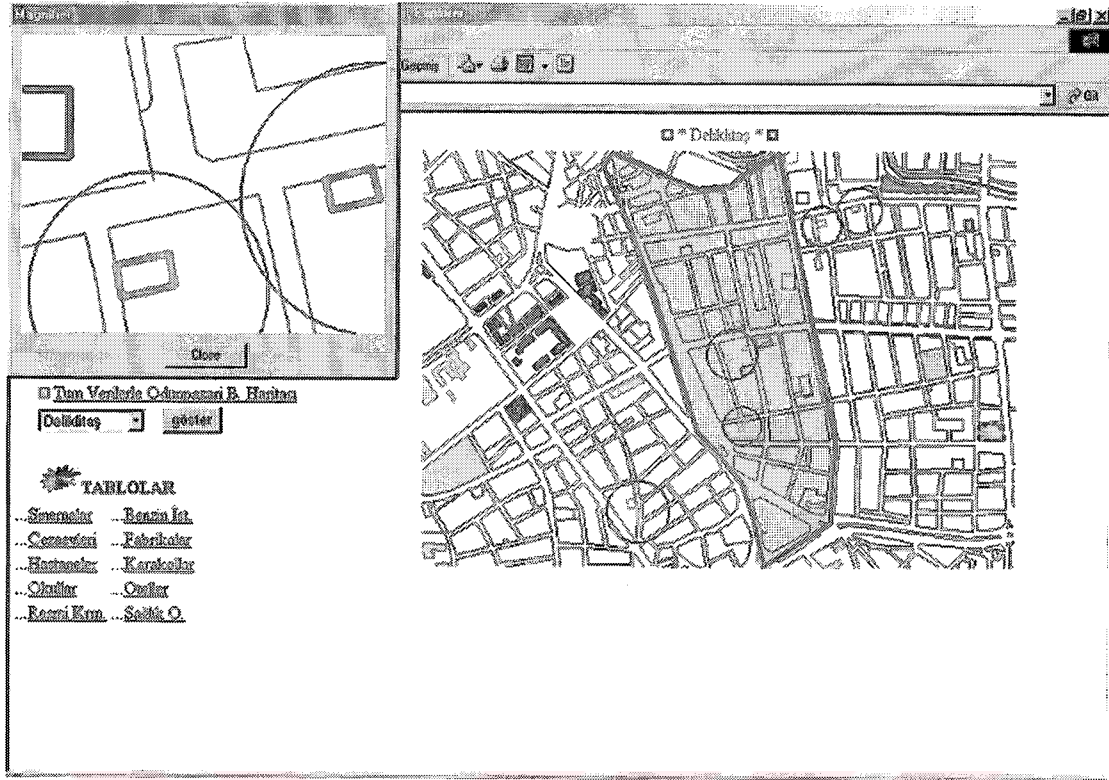
Şekil 5.9 Harita üzerinde işaretleme yapmak üzere kullanılan menü

Şekil 5.10’da ise, harita üzerinde kullanıcı işlemleri görülmektedir.

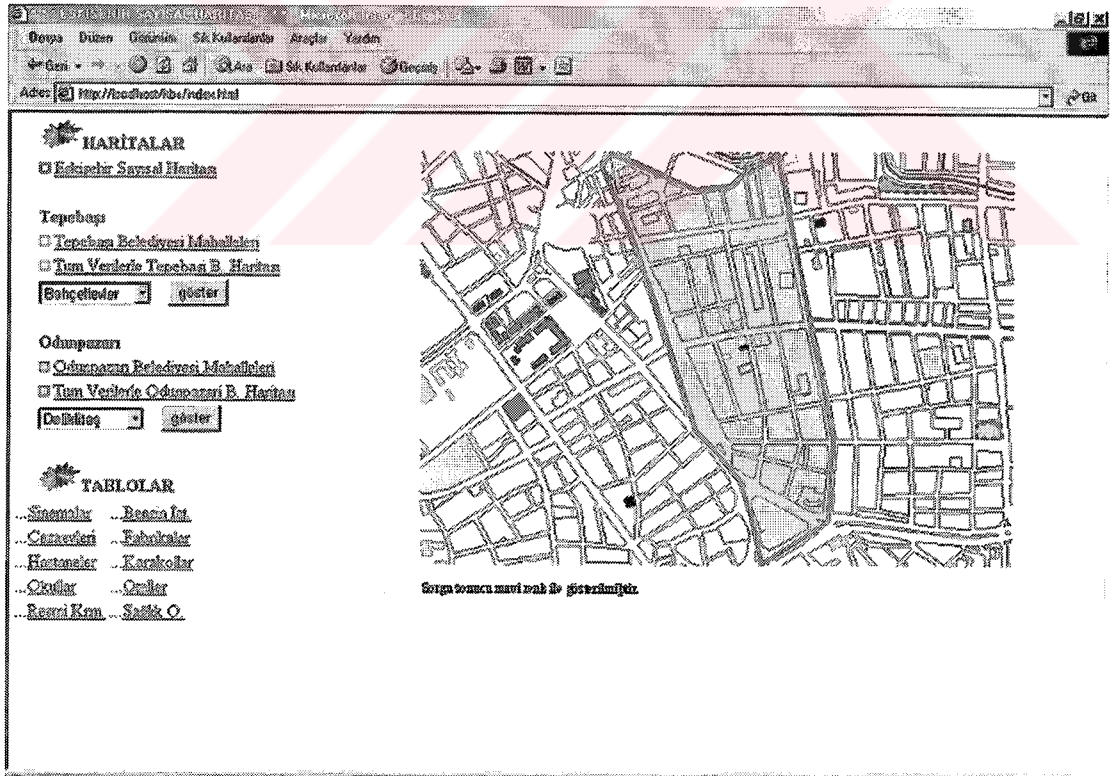


Şekil 5.10 Ziyaretçinin harita üzerinde çizim ve yazım ile işaretleme yapması

Ayrıca ASP kodları kullanılarak harita üzerinde sorgulamalar yapılabilir. Şekil 5.11’de, Deliklitaş Mahallesi’ndeki öğrenci yurtları görülmektedir. ASP kodları ile yapılabilecek bir sorgulama ile yurtlar ile ilgili girilmiş veriler taranarak sorgulama yapılabilir. Örneğin, yurtların kız öğrenci yurdu mu, erkek öğrenci yurdu mu olduğu bilgisinin girildiği sütun taranarak, sadece erkek öğrenci yurtlarını göster gibi bir sorgulama yapılabilir. Şekil 5.12’de sorgu sonuçları gösterilmiştir.



Şekil 5.11 Deliktaş Mahallesi civarındaki öğrenci yurtları



Şekil 5.12 Sorgu sonucu erkek öğrenci yurtlarının gösterilmesi

Bütün bu harita üretiminin arkasındaki ASP dosyaları şu yapıdadır:

Öncelikle ASP kodları, HTML kodları arasına yerleştirilmiştir.

```
<html>
<head>
<meta name="generator" content="microsoft visual studio 6.0">
<title>KENT BİLGİ SİSTEMİ</title>
</head>
<body>

[BURADA ASP KODLARI YER ALIR]

<center><font color=rgb(220,18,132)>
...<font size=4>ESKİŞEHİR</font>...
<br></center>
<center>
<font color=rgb(79,153,253)>..ODUNPAZARI
BELEDİYESİ
<font color=rgb(228,145,0)>..TEPEBAŞI BELEDİYESİ
<br><font color=red>..İl Sınırı,
<font color=rgb(4,170,109)>..Porsuk Nehri,
<font color=rgb(197,6,136)>..Demiryolları<br>
</font></center>
</body>
</html>
```

Görüldüğü gibi HTML sayfası açıldıktan sonra başlık tanımlanmış, ancak gövde kısmında harita üretimini sağlayan ASP kodları kullanılmıştır. ASP kodları bitirildikten sonra yine HTML kodları ile kapanmıştır.

Arada <% ve %> kodları arasına yazılan ASP kodları ise şu şekilde hazırlanır:
Öncelikle,

```
<%@LANGUAGE="VBSCRIPT"%>
```

komutu ile vbscript dili kullanılacağı belirtilir.

Daha sonra veritabanına ulaşmayı sağlayan asıl komutlar gelir. Koordinat sistemi bilgilerini içeren utm36.csf isimli dosya tanıtılarak, koordinat sistemi ile ilgili bilginin buradan alınacağı belirtilir. Hemen arkasından proje klasöründe veritabanının yeri, veritabanının MS Access dosyası olduğu belirtilir ve bir bağlantı adı verilir. Bundan sonra da verilerin girildikleri ölçekte (x1) görüntülenmek istendiği sisteme bildirilmiştir.

```
<%  
Set objSesMgr = Server.CreateObject("GMWebMap.MapServerManager")  
'-----  
Set objMS = objSesMgr.MapServer("")  
objMS.Clear  
objMS.OutputType = 0  
objMS.SetCoordinateSystem "C:\inetpub\wwwroot\kbs\utm36.csf"  
objMS.Connect "Access.GDatabase",  
"C:\INETPUB\WWWROOT\kbs\kent.mdb", "", "Baglanti1"  
  
Set oAS1 = objMS.AddRangeAutoScale("oAS1")  
oAS1.Multiplier = 1
```

Bu bilgilerin ardından sistem veritabanına ulaşır, verilerin geometrik bilgilerini de ilgili sütundan alır ve kendisine tanıtılan koordinat sistemi bilgilerini de kullanarak, haritayı üretmeye hazır hale gelir. Ancak ulaştığı veritabanında hangi tabloların gösterilmek istendiği ve bu tablolarda kayıtları tutula coğrafi nesnelerin ne şekilde gösterileceği belirtilmelidir.

Örneğin Şekil 5.1’de gösterilen genel Eskişehir haritası için hazırlanan haritada, veritabanındaki Tepebaşı ve Odunpazarı tablolarının kullanılacağı aşağıdaki komutlar ile sağlandığı gibi, gösterim şekli, rengi gibi özellikler de belirtilir.

```
Dim Cols(1)  
Cols(0) = "ID"  
Cols(1) = "Mahalle_Adi"  
  
Set oMkr = objMS.AddMarker("odm", "Baglanti1", "odunpazarı", "", Cols)  
oMkr.AddElements ""  
'-----  
Set oDisplayRule = oMkr.NewDisplayRule  
  
Set oSym = objMS.CreateObject("GMWebMap.GWMFeatureSymbology")  
  
oSym.Color = rgb(79,153,253)  
oSym.Weight = 2
```

```

oSym.FillColor = rgb(202,223,253)
oSym.Style=0

oDisplayRule.DisplaySymbology = oSym
'oDisplayRule.HotspotSymbology = oSym

Set oSym = Nothing
oDisplayRule.ToolTip = "<" & "%" & ".Mahalle_Adi" & "%" & ">"

'oDisplayRule.Action = "http://localhost/esk/dokuman.asp?BT=<%Adi%" & ">"

oMBR = oMkr.Range
'-----
    Dim aCols(1)
    aCols(0) = "ID"
    aCols(1) = "Mahalle_Adi"

    Set oMkr = objMS.AddMarker("tpm", "Baglanti1", "tepebasi", "", aCols)
    oMkr.AddElements ""
'-----
    Set oDisplayRule = oMkr.NewDisplayRule

    Set oSym = objMS.CreateObject("GMWebMap.GWMFeatureSymbology")

    oSym.Color = rgb(254,179,48)
    oSym.Weight = 2
    oSym.FillColor = rgb(251,246,213)
    oSym.Style=0

    oDisplayRule.DisplaySymbology = oSym
'oDisplayRule.HotspotSymbology = oSym

    Set oSym = Nothing
    oDisplayRule.ToolTip = "<" & "%" & ".Mahalle_Adi" & "%" & ">"

'oDisplayRule.Action = "http://localhost/esk/dokuman.asp?BT=<%Adi%" & ">"

oMBR = oMkr.Range

```

Burada aynı zamanda tabloyu oluşturan mahalleler ile ilgili, mahalle adı sütununa gidilerek, fare nesnenin üzerine gelince gösterilecek bilgi işaret edilmektedir.

Porsuk Nehri veya demiryolu gibi tek parça veriden oluşan ya da öz nitelik bilgisi girilmemiş verilerin ASP kodları ile sisteme tanıtılması biraz daha kolaydır:

```

Set DisplayRule = objMS.AddGeomColumn ("Baglanti1","porsuk","")

Set Sym = objMS.CreateObject("GMWebMap.GWMFeatureSymbology")
Sym.Color = RGB(4,170,109)
Sym.Weight = 2
Sym.Style = 0

DisplayRule.DisplaySymbology = Sym

'DisplayRule.HotspotSymbology = Sym

Set Sym = Nothing
DisplayRule.ToolTip = "%" & ".porsuk nehri." & "%"

```

```

Set DisplayRule = objMS.AddGeomColumn ("Baglanti1","demiryolu","")

Set Sym = objMS.CreateObject("GMWebMap.GWMFeatureSymbology")
Sym.Color = RGB(197,6,136)
Sym.Weight = 1
Sym.Style = 0

DisplayRule.DisplaySymbology = Sym

Set Sym=Nothing
DisplayRule.ToolTip= "%" & ".demiryolu." & "%"

```

Web sitesinin sol tarafında yer alan drop down menü, şu komutlar ile düzenlenmiştir:

```

<table>

<Form action="bult.asp" method="POST" target="asg">
  <SELECT NAME="mah" SIZE="1">

    <OPTION SELECTED VALUE="bahcelievler">Bahçelievler
    <OPTION VALUE="cumhuriye">Cumhuriye
    <OPTION VALUE="camlica">Çamlıca
    <OPTION VALUE="ertugrulgazi">Ertuğrul Gazi
    <OPTION VALUE="esentepe">Esentepe
    <OPTION VALUE="eskibaglar">Eskibağlar
    <OPTION VALUE="fatih">Fatih
    <OPTION VALUE="fevzicakmak">Fevzi Çakmak
    <OPTION VALUE="gulluk">Güllük
    <OPTION VALUE="hacialibey">Hacı Ali Bey
    <OPTION VALUE="haciseyit">Hacı Seyit
    <OPTION VALUE="hayriye">Hayriye
    <OPTION VALUE="hosnudiye">Hoşnudiye
    <OPTION VALUE="isiklar">Işıklar

```



```

elseif instr(mhl, "seyrantepe") <> 0 THEN
    gonder= "sy.asp"
.....
.....
.....
elseif instr(mhl, "seker") <> 0 THEN
    gonder= "skr.asp"
elseif instr(mhl, "fevzicakmak") <> 0 THEN
    gonder= "fc.asp"
else gonder="yok.html"
end if
Response.Redirect gonder

%>

```

Bu durumda, dropdown menüde yapılan seçimin anahtar kelimesi alınıp bult.asp'ye yönlendirilmek üzere gelinmiş olur. Arka planda çok kısa bir sürede tamamlanan bu işlem sonrası örneğin seçilen mahalle fevzicakmak ise, fc.asp çalıştırılmaktadır.

fc.asp'de Eskişehir'in genel haritasının üretildiği .asp gibi oluşturulmuştur. Öncelikle veritabanı gösterilir, daha sonrada buradan alınacak veriler komutlar ile belirtilir.

Sorgu asp'lerinde ise, öncelikle, hangi sütunun taranacağı belirtilir. Daha sonra where kalıbı kullanılarak, sorgu sonucu yayınlanır.

Optimizasyon ve Performans Gerekleri, Sistem Güvenliği:

Elemanları tıklanabilir hazırlamak ve farklı eleman gruplarına özgü bazı fonksiyonlar eklemek dosya boyutlarını arttıracaktır. Büyük boyutlu dosyalar, üretilen haritaların istemci tarafa yüklenmesini yavaşlatacaktır.

Yüklemede performansı arttırmak için ACGM dosyası boyutlarını küçültmek gerekir. ACGM ürününü optimize etmek için aşağıdaki tavsiyelere uymak performansı arttıracaktır:

Veri gruplarından birinin elemanları üzerine mouse ile gelince, ilgili bir yazı gösterilecek olsun. Bu durumda belirecek yazıda mümkün olduğunca az kelime olmasına çalışılır, hatta mümkünse tanıtıcı amaçlı tek bir kelimeyi geçmemelidir.

Sunucu üzerinde çalışan ASP ve HTML dosyalarına (ayrıca proje klasöründe bulunan tüm dosyalara) ve istemci makinada çalışacak JavaScript fonksiyonlarına kısa isimler verilmelidir.

İyi dizayn edilmiş, index kullanılmış bir veri tabanı hızı arttıracaktır.

Sistem güvenliği açısından fire wall kullanılması veya profesyonel ve büyük çaplı çalışmalarda verinin sunucu makinada değil, uzak bir sunucu üzerinde bulunması güvenlik açısından daha iyi olacaktır.



6. SONUÇLAR

Coğrafi Bilgi Sistemleri 1980'lerden sonra, gelişmiş dünya ülkelerinde kullanılmaya başlanmış ve bu konuda ülkemizde de bir çok kurum tarafından projelere başlanmıştır. Bu projeler genellikle kurumların kendi konularıyla ilgili bilgi sistemlerini kapsar niteliktedir. Böyle çalışmalarda tüm kurum ve kuruluşların katılımı ve işbirliği gerçekleştirmesi ve projede yer alması, öngörülen detay kodlama kataloğu, veri sözlüğü gibi standartların belirlenmesi gerekmektedir.

Öncelikle kentlerimiz için yapılandırılacak kent bilgi sistemlerine ihtiyaç vardır. Kent yönetimleri ve buna bağlı kurum ve kuruluşların; hem kendi içlerinde, hem de birbirleri arasında bilgi kullanımı ve paylaşımını yeni teknolojik olanaklardan yararlanarak sağlamaları, kazanılan avantajların yanı sıra mevcut sistemlerdeki zaman, para ve işgücü kaybını önleyecektir. Kent yönetimi ve ilgili kuruluşlar için Kent Bilgi Sistemleri kaçınılmaz bir teknolojik gelişmedir.

Kent Bilgi Sistemleri, coğrafi bilgi sistemleri ve veritabanı teknolojisinin bir arada kullanılması ile oluşturulur. Etkin kullanım için, veritabanı tasarımının en iyi şekilde yapılması gerekmektedir. Bu amaçla, tez kapsamında, ilk önce CBS ve veritabanı teknolojileri incelenmiş ve daha sonra, bu iki teknoloji kullanılarak oluşturulacak olan kent bilgi sistemleri ele alınmıştır. CBS sistemlerinin veritabanı sistemlerinden farklı olarak coğrafi verideki grafik bileşenini görsel olarak ifade edebilmesi ve katman mantığı ile ortaya koyması onu en kullanışlı hale getiren özelliğidir, ve bilgisayar ortamının güncelleme kolaylığı hızlı öngörü gibi, veri nnnnnnnnnn üzerinde analiz yapılabilir olması en önemli avantajlarından biridir.

Tezde, bir kent bilgi sisteminin içermesi gerekenler ve veritabanı oluştururken dikkat edilecek hususlar ele alınmıştır. Bu tez kapsamında ağırlıklı olarak kent yönetimi amaçlı,

detaylı KBS sistemlerinin dışında; internet üzerinden yayınlanmak üzere genel bilgi içeren ve çok sayıda kişinin paylaşımına açık olan diğer bir kullanım şekli incelenmiştir.

Gelişmiş toplumlarda kullanılan CBS'nin çok sayıda kişinin kullanımına ve sorgulamasına açık olan internet uygulaması örnekleri incelenmiştir. Bu uygulamalar incelendiğinde çok detaylı çalışmalardan, çok genel çalışmalara kadar pek çok örnek bulunur. Pek çoğunda sağlanan sorgulama imkanının gelecekte ilişkisel veritabanı mantığı kullanılarak çok daha esnek olması planlanmaktadır.

Tez kapsamında yapılan uygulamada, Eskişehir'e ait bir kent bilgi sistemi ve internet üzerinden yayınlanması projesi hazırlanmıştır. Bu uygulamada ASP kodları kullanılarak her seferinde istemci makinadan gelen isteğe bağlı olarak yeni harita üretilmektedir.

İnsanların çoğu ihtiyaçlarını internet üzerinden sağlamaları çok yakın gelecekte gerçekleşecek ve Kent Bilgi Sistemleri de insanların yaşadıkları kentler ile ilgili pek çok bilgiye evlerinden ulaşabilecekleri etkileşimli haritalar olarak internet dünyasında yerlerini alacaklardır.

KAYNAKLAR

- [1] : Ana Britannica Cilt 14, s. 431
- [2] : **Rhind, D.**, 1993, Mapping for the New Milennium, 16. International Cartographic Conference, Cologne, s. 3-14
- [3] : Ana Britannica, Cilt 14, s. 432
- [4] : **Kraak, M. and Ormelling, F.**, 1996. Cartography: Visualization of Spatial Data, Longman
- [5] : **Batuk, G., Külür, S., Sarbanoğlu H., Toz, G.**, 1996, İstanbul, Veriden Bilgiye: Coğrafi Bilgi Sistemleri, Coğrafi Bilgi Sistemleri Sempozyumu
- [6] : **Akgündüz, E.**, 1997, Coğrafi Bilgi Sistemi Uygulamalarında Ülkemizdeki Durum, *Yüksek Lisans Tezi*, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü
- [7] : **Alkış, Z.**, 1996, Yönetimlerde Karar-Destek Sistemi Olarak Coğrafi Bilgi Sistemlerinin Önemi, Coğrafi Bilgi Sistemleri Sempozyumu 96
- [8] : **Altan, M. O., Toz, F. G., Külür, S.**, 1996, İTÜ Bilgi Sistemlerindeki Gelişmeler ve Fotogrametri, Coğrafi Bilgi Sistemleri Sempozyumu, İstanbul
- [9] : **Zeren, B.**, 2000, Uçuş Planlama Sistemi Tasarımı, *Bitirme Tezi*, İTÜ, Uçak ve Uzay Bilimleri Fakültesi
- [10] : **Silberschatz, A., Korth, H. F., Sudarshan S.**, Database System Concepts, McGraw Hill International Editions
- [11] : **Thomas, C., Begg, C., Strachan, A.**, 1996, Database Systems, A Practical Approach to Design, Implementation and Management, Addison –Wesley Publishing Company,
- [12] : **Elmasri, R., Navathe, S. B.**, 1994, Fundamentals of Database Systems, The Benjamin Cumming Publishing Company Inc.
- [13] : **Çetiner G.**, Veri Modelleme Notları
- [14] : **Lorents, A.C., Morgan, J. N.**, 1998, Database Systems, Concepts, Management and Applications, The Dryden Press
- [15] : **Çetiner G.**, SQL Ders Notları
- [16] : www.islem.com.tr/altsayfalar/islem/uygulamayaz/kentbilgisistemi.htm
- [17] : www.hkmo.org.tr/yayin/mulkiyet/7cis.htm:
Coğrafi Bilgi Sistemleri, Mehmet Dikici, Kadastro Müdürü
- [18] : **Newell, R. G., Theriault, D. G.**, 1990, Is GIS just a Combination of CAD and DBMS, Mapping Awareness, vol4, no3, pp42
- [19] : **Maguire, D. J., Goodchild, M. F., Rhind, D. W.**, 1991, Geographical Information Systems Principles and Applications, Longman.
- [20] : **Güngör, V.**, 1999, Coğrafi Bilgi Sistemlerinde Ağ Analizi, *Yüksek Lisans Tezi*, İTÜ, Fen Bilimleri Enstitüsü
- [21] : **Thomas, B. J.**, Spie Optical Engineering Press, 1997, The Internet For Scientists and Engineers,
- [22] : **Gür, Ö.**, 2000, İnternet Üzerinden Reklamcılık, *Yüksek Lisans Tezi*, Anadolu Üniversitesi, İletişim Bilimleri Fakültesi, Eskişehir

- [23] : **Tarhan, C.**, 1999, Webmaster için El Kitabı, Pusula Yayıncılık ve İletişim Ltd.
- [24] : **Öcal, H.**, JavaScript ile Dinamik Web Sayfaları, 1999, İhlas Gazetecilik Yayıncılık
- [25] : **Öcal, H.**, Active Server Pages Programcılarını için Temel ASP Rehberi (I), 2000, İhlas Gazetecilik Yayıncılık
- [26] : **Karahoca, A.**, ASP ile Web Programcılığına Giriş, 1999, Beta B.Yayın.Dğt.



ÖZGEÇMİŞ

17.05.1976 yılında Eskişehir’de doğdu. Murat Atılgan İlkokulu’nda başladığı öğrenimine Eskişehir Anadolu Lisesi’nde devam etti. 1994 yılında İstanbul Teknik Üniversitesi Uçak ve Uzay Bilimleri Fakültesi, Uzay Bilimleri ve Teknolojisi Mühendisliği Bölümüne girdi. 1998 senesinde bu bölümden mezun oldu. Daha sonra İstanbul Teknik Üniversitesi, Bilişim Teknolojileri yüksek lisans programına girmeye hak kazandı.



**İC TİPİSİ ÖĞRETİM KURULU
DOKÜMENTASYON MERKEZİ**