

**COĞRAFİ BİLGİ SİSTEMLERİNDE  
NESNEYE DAYALI VERİ MODELLEMESİ**

**YÜKSEK LİSANS TEZİ**  
**Jeodezi ve Fotogrametri Mh. Lütfiye KUŞAK**  
**(501991080)**

**Tezin Enstitüye Verildiği Tarih: 10 Mayıs 2002**  
**Tezin Savunulduğu Tarih: 5 Haziran 2002**

**Tez Danışmanı : Prof. Dr. Doğan UÇAR**  
**Diğer Jüri Üyeleri Prof. Dr. Mehmet SELÇUK (Y. T. Ü)**  
**Doç. Dr. Necla ULUĞTEKİN (İ. T. Ü)**

**HAZİRAN 2002**

## ÖNSÖZ

İ. T. Ü. Fen Bilimleri Enstitüsü, Jeodezi ve Fotogrametri Mühendisliği Anabilim Dalı Jeodezi ve Fotogrametri Mühendisliği Program'nda gerçekleştirilen bu yüksek lisans çalışmasında Coğrafi Bilgi Sistemlerinde nesneye dayalı veri modellemesi kavramı ve uygulaması incelenmiştir.

Öncelikle çalışmalarım esnasında tez danışmanlığımı üstlenen, her aşamada geniş bilgi birikimi ve yapıcı eleştirileri ile beni yönlendiren ve sürekli yeni sonuçlar üretmeye zorlayan sayın hocam Prof. Dr. Doğan UÇAR'a teşekkür etmeyi isterim. Ayrıca bana her konuda destek olan Doç. Dr. Cengizhan İPBÜKER ve Doç. Dr. Necla ULUĞTEKİN başta olmak üzere tüm Anabilim Dalı öğretim elemanlarına teşekkür ederim.

Son olarak bana maddi ve manevi açıdan her türlü desteği veren, her konuda anlayış gösteren bu çalışmayı ortaya koymada büyük emekleri geçen babam Mehmet KUŞAK'a, annem Hilal KUŞAK'a ve kardeşim Bahriye KUŞAK'a sonsuz teşekkürlerimi sunarım.

HAZİRAN 2002

Lütfiye KUŞAK

## İÇİNDEKİLER

<b>KISALTMALAR</b>	<b>vi</b>
<b>TABLO LİSTESİ</b>	<b>vii</b>
<b>ŞEKİL LİSTESİ</b>	<b>viii</b>
<b>ÖZET</b>	<b>ix</b>
<b>SUMMARY</b>	<b>x</b>
<b>1. GİRİŞ ve CBS</b>	<b>1</b>
1.1. Bilgi Sistemleri Nedir?	2
1.2. CBS Tanımı	2
1.2.1. CBS kullanımları	3
1.3. CBS’de Kullanılan Veri	4
1.3.1. Mekansal Verinin Genel Özellikleri	5
1.3.2. Mekanın modellenmesinde kullanılan veri kavramları	6
1.3.2.1. Konumsal veri	6
1.3.2.2. Öznitelik verisi	6
1.3.2.3. Zaman	6
1.3.2.4. Meta data	6
1.4. CBS’nin Fonksiyonel Bileşenleri	6
1.4.1. Veri aktarımı	6
1.4.2. Veri depolama	7
1.4.3. Veri işleme	7
1.4.4. Mekansal analiz	7
1.4.5. Sorgulama	7
1.4.6. Veri sunma	7
1.5. CBS’nin Yararları	8
1.6. Türkiye’de CBS ve Uygulamaları	8
1.6.1. Merkezi idareler	9
1.6.2. Mahalli idareler	9
1.6.3. Özel sektör	9
<b>2. MODELLEME DİLLERİ</b>	<b>11</b>
2.1. Yapısal Modelleme	12
2.2. Nesneye Dayalı Yaklaşımlar	14
2.3. Nesneye Dayalı Programlamanın Temel Amacı	14
2.4. C++ Dili	15
2.5. Nesneye Dayalı Yaklaşımın Temelleri	15
2.6. Nesneye Dayalı Programlamada Kullanılan Nesne	16

2.7. Nesneye Dayalı Veri Modellemesinin Özellikleri	17
2.7.1. Objelerin sınıfları	18
2.7.1.1. Objelerin sınıflarında kullanılan operasyonlar	21
2.7.2. Objeler arasındaki ilişkilerin kurulması	23
2.7.2.1. Agregasyon	23
2.7.2.2. Asosiyasyon	24
2.7.3. Objelerin kimliği	25
2.7.4. Depolama (encapsulasyon)	26
2.7.5. Otomatik öznitelik ve fonksiyon üstlenme (miras)	28
2.7.5.1. Tekil miras	29
2.7.5.2. Çoklu miras	30
2.7.6. Çok biçimlilik (polimorfizm)	32
2.7.7. Yöntemleri farklılaştırma (overriding)	34
2.7.8. Nesne bildirme (overloading)	34
<b>3. VERİ TABANI YAPILARI VE VERİ TABANI YÖNETİMİ</b>	<b>35</b>
3.1. Hiyerarşik Veri Tabanı Yapıları	36
3.2. Ağ Veri Tabanı	37
3.3. İlişkisel Veri Tabanı	37
3.4. Nesneye Dayalı Veri Tabanı	39
3.5. Genişletme Kavramı	42
3.5.1. İlişkisel veri tabanları	43
3.5.2. En uygun veri tabanı yapısının seçimi	43
3.6. Veri Tabanı Yönetim Sistemleri	44
3.6.1. Vektör ve raster veri yapılarının veri tabanında saklanması	45
3.6.2. Nesneye dayalı veri tabanı yapıları: birleştirilen öznitelikler ve geometrik saklama	46
3.6.3. Nesneye dayalı veri modellemesi yönetim sistemlerinin özelliklerini coğrafi obje yönetimiindeki faydaları	47
3.6.4. İlişkisel hibrit CBS'ler ile nesneye dayalı yöntemlerin karşılaştırılması	48
3.7. CBS Amaçlı Sorgulama Dilinde Bulunması Gereken Özellikler	50
<b>4. SUPPORT GIS YAZILIMI</b>	<b>52</b>
4.1. Support GIS'in Temel Yapısı	52
4.1.1. Support GIS de kullanılan şekiller	53
4.1.2. Support GIS de kullanılan sorgulama	58
4.1.3. Support GIS de grafik işlemleri	67
4.2. Support GIS'in Genel Özellikleri	68
4.3. Support GIS de Uygulanan İşlemler	69
4.3.1. Sınıflara ait özniteliklerin oluşturulması	69

<b>5. SONUÇLAR VE ÖNERİLER</b>	<b>74</b>
<b>KAYNAKLAR</b>	<b>76</b>
<b>EKLER</b>	<b>79</b>
<b>ÖZGEÇMİŞ</b>	<b>93</b>

## KISALTMALAR

**OOP** : **O**bject-**O**riented **P**rogrammi ng  
**ADT** : **A**bstract **D**ata **T**ype  
**CBS** : **C**oğrafi **B**lgi **S**temi  
**SQL** : **S**tructured **Q**uery **L**anguage  
**TAKBİS** : **T**apu **K**adastro **B**lgi **S**temeri

## TABLO LİSTESİ

	<b>Sayfa No</b>
<b>Tablo A 1</b> : Ada sınıfına ait Öznitelik verileri	79
<b>Tablo A 2</b> : B na sınıfına ait Öznitelik verileri tablosu	80
<b>Tablo A 3</b> : M lık sınıfına ait öznitelik özellikleri tablosu	81
<b>Tablo A 4</b> : Şahıs sınıfına ait öznitelik verileri	82
<b>Tablo A 5</b> : Parsel sınıfına ait özniteliklerin gösterimi	83
<b>Tablo G 1</b> : Hasköy Bölgesine ait m lık bilgileri	89
<b>Tablo G 2</b> : Hasköy Bölgesi kişilere ait parsel cilt no, yevmiye, edinme şekli ve tarih bilgileri	90

## ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1 : Parsel ve sokakların gösterildiği bir kadastro haritası	13
Şekil 2.2 : Yapısal modelleme ile yapılmış bir kadastro işlemi	13
Şekil 3.1 : Hiyerarşik veri modeli	36
Şekil 3.2 : Ağ veri modeli	37
Şekil 3.3 : İlişkisel veri modeli	38
Şekil 3.4 : İlişkisel veri modeli gösterimi	40
Şekil 3.5 : Nesneye dayalı veri modeline göre aynı işlemin gösterilmesi	40
Şekil 3.6 : Nesneye dayalı veri modellemesine ait bir gösterim	41
Şekil 4.1.a: Support G S' de çizgi objelerinin oluşturulması	56
Şekil 4.1.b: Çizgi objeleri oluşturacak olan noktaların belirlenmesi	56
Şekil 4.2 : Support G S' de yazı melekmanlarının girilmesi	57
Şekil 4.3 : Sınıflar ve çizimobjeleri arasındaki ilişkileri tanımlanması	57
Şekil 4.4 : Support G S' de kullanıcı yönetimi	58
Şekil 4.5 : Support G S' de sorgulama işleminde düğüm ve kenarların gösterilmesi	59
Şekil 4.6 : Öznitelikler kullanılarak yapılan sorgulama işlemi	59
Şekil 4.7 : Sınıflar arasındaki ilişkilerin sorgulanması işlemi	60
Şekil 4.8 : Karşı sorgulama işlemi	61
Şekil 4.9 : Sınıf düğümlerinin oluşturulması	62
Şekil 4.10 : Seçme düğümlerinin oluşturulması	63
Şekil 4.11 : Objesınıfları ve şahıssınıfı arasındaki değerlerin oluşturulması	65
Şekil 4.12 : Sınıflara ait oluşturulan ID numaraları	66
Şekil 4.13 : Oluşturulan ID değerlerine karşılık gelen öznitelik gösterimi	66
Şekil 4.14 : ID değerlerine karşılık gelen ilişkilerin gösterimi	67
Şekil 4.15 : Sorgulama işlemlerinin taranması	68
Şekil B.1 : Support G S' de sınıf gösterimi	84
Şekil B.2 : Support G S' de öznitelik verilerinin oluşturulması	85
Şekil B.3 : Support G S' de sınıflar arasındaki ilişkilerin kurulması	86
Şekil B.4 : Bna ve parsele ait verilerin oluşturulması	87
Şekil B.5 : Sorgulama sonuçlarının gösterilmesi	88
Şekil G.3 : Hasköy bölgesi için uygulama çalışmaları	91



## ÖZET

Günümüzde, bilgi teknolojileri insan yaşam için çok önemli bir araç haline gelmiştir. Bilgi sistemleri bilginin yönetilmesinde kullanılırlar. Bu çalışmada mekansal verinin işlenmesinde kullanılan Coğrafi Bilgi Sistemi (CBS) incelenmiştir.

Coğrafi Bilgi Sistemi, gelişmiş ülkelerde kadaströ, planlama gibi farklı alanlarda mekansal verinin işlenmesinde kullanılır. CBS Kamu kurumları, özel sektörler ve yönetimler için, çok daha hızlı, güvenilir, az tekrar edilen ve çok daha az zaman harcayacak şekilde çalışmalar sunmaktadır. Bu çalışmada mekansal verinin saklanması ve işlenmesi için CBS’de kullanılan veri modellerinin incelenmesi amaçlanmıştır.

Çalışmanın ilk bölümünde, CBS ve CBS’ nin yararları ve uygulama alanları hakkında genel bilgi verilmiştir.

Çalışmanın ikinci kısmında, CBS’de kullanılan veri modelleri anlatılmıştır. Tez konusu olan CBS’ de nesneye dayalı veri modellemesi hakkında bilgi verilmiş ve ilişkisel veri modeli ile karşılaştırılmıştır.

Sonuncu bölümünde, Bonn üniversitesi çalışanları tarafından nesneye dayalı programlama dili kullanılarak geliştirilen Support GS yazılımı hakkında bilgi verilmiştir. Çalışmada Support GS yazılımında nesneye dayalı veri modellemesi özellikleri incelenmiştir. Yazılımbir pilot bölgede kadaströ bilgi sistemi geliştirilerek denenmiştir. Çalışma bölgesi Beyoğlu Belediyesi ne ait Hasköy Bölgesinde yapılmıştır. Sonuçlar elde edilmiş ve bir takım öneriler getirilmiştir.

Çalışmanın sonunda, mekansal verinin yönetilmesinde CBS’de nesneye dayalı veri modellemesi sistemlerinin ilişkisel veri modellemesine göre daha başarılı olduğu görülmektedir.

## **OBJECT-ORIENTED MODELING IN GEOGRAPHICAL INFORMATION SYSTEMS**

### **SUMMARY**

No wadays, information technology is being a very important tool for human life. Information systems are used to managing information. In this study, Geographical Information System (GIS) used for manipulation of spatial data is examined.

GIS are used especially in developing countries to manipulate of spatial data in different areas, such as cadastre, land planning etc. GIS present faster, more successful, more reliable, less unrepeat and less spending works for municipality, private enterprise and government. In this study, it is aimed to examine kinds of modelling systems in GIS to store and manipulate of spatial data.

At the first stage of this study, a general definition is given for Geographical Information Systems and its application area and benefits.

At the second part of the study, data models used in GIS are explained. Especially information about object-oriented data modeling system in GIS, being subject of the thesis project, is given and compared to relational data modeling for the GIS.

In the last chapter, general information about the software namely Support GIS is given. Support GIS software has been developed using an object oriented program language by a staff from Bonn University. Properties of object-oriented modeling systems are examined to Support GIS software in this study. Support GIS software is tried to develop a cadastre information system as a small size pilot project. The study area includes the Region of Hasköy, Beyoğlu Municipality. Finally the results came out and due to these results some suggestions are made.

At the end of the study, it is mentioned that the object-oriented data modeling systems seem more successful than relational modeling systems for managing spatial data in GIS.

## 1. GİRİŞ

20. yüzyılın özellikle ikinci yarısında gelişen bilgisayar teknolojisi bilgiyi ve bilgini kullanmanın daha önemli hale getirmiştir. Bugün gelişmiş olan pek çok ülke elde edilen verilerin en doğru ve güvenilir biçimde tutulmasının yansınası verilerin en uygun şekilde kullanılmasını sağlayan CBS (Coğrafi Bilgi Sistemi) kullanmaktadır. CBS iş verimliliğini ve başarısını artırması, bilgi akışını hızlandırması, mevcut verilerden analiz yapabilmesi, güncelleştirmenin kolaylıkla yapılabilir olması, iş gücü ve zaman kaybını önlemesi nedeniyle dünyada Kamu Kurum ve Kuruluşlarında ve özel sektörlerde özellikle tercih edilmektedir. Ülkemizde ise çalışmalar olmakla birlikte, kurumların birbirinden bağımsız kendi olanakları ile işlem yapmalarından dolayı sorunlar içermektedir.

Yapılan çalışmanın birinci bölümünde adı geçmekte olan CBS hakkında bilgi verilmiştir. Türkiye’de ve genel olarak kullanım alanları incelenmiş ve yararları üzerinde durulmuştur.

Çalışmanın ikinci kısmında ise gelişen teknolojinin getirmiş olduğu bazı kolaylıkların CBS’ye yansınası incelenmiştir. Bunun sonucunda CBS için geliştirilmiş olan bazı veri modelleri ve özellikle tez konusu olan “*nesneye dayalı veri modelleri*” hakkında bilgi verilmiştir. Nesneye dayalı veri modellerinin içerdiği miraslanma, sınıf özellikleri, veri depolanması, çok biçimlilik konuları hakkında bilgi verilmiş ve bu yapıların CBS’de kullanım alanları açıklanmıştır.

Çalışmanın sonuncu bölümünde İTÜ İnşaat Fakültesi bünyesinde Federal Almanya Cumhuriyeti Bonn Üniversitesi Kartografya ve Geoinformasyon Enstitüsü Jeodezi ve Fotogrametri Mühendisleri ve Bilgisayar Mühendisleri tarafından geliştirilen bir CBS yazılımı olan Support GS incelenmiştir. Çalışmada mekansal objelerin değerlendirilmesinde en doğru model olan nesneye dayalı yazılım kullanılarak geliştirilen Support GS ve özellikleri belirlenmiş ve Beyoğlu Belediyesi Hasköy

Bölgesi İmar çalışması üzerinde denenmiştir. Support GIS'in veri girişi ve sorgulama yapısı kavramları açıklanmıştır.

Support GIS sayesinde obje sınıfları oluşturulmuş ve sınıflara ait öznelik ve fonksiyonlar atanmıştır ve aralarındaki ilişkiler tanımlanmıştır. Sorgulama dili sayesinde mekansal veri türlerine ilişkin sorgulamalar, mantıksal operasyonlar uygulanabilmiş, böylece kompleks sorgulamalar formüle edilmiştir. Sistemin özelliği sayesinde kullanıcıya her aşamada bir uzman ile çalışıyormuş duygusu hissettirilmiş böylece uygulamada hata yapma olanağı azaltılmıştır. Support GIS yazılım günümüzde internet ortamında faaliyeti sürdürmektedir.

### **1.1. Bilgi Sistemleri Nedir?**

Artan nüfusun beraberinde getirdiği hizmetlere olan talep artışının kontrol edilemezliği Merkezi İdareleri ve Yerel Yönetimleri hareket edemez hale getirmektedir. Hizmetlere ait gereksinimlerin saptanması imkansızlaşmaktadır. Planlama ve kontrolün ortadan kalmasına da neden olmaktadır. Bunun sonucunda hizmet değerleri elde edilememekte, hemde maksimum düzeyde gerekli olan kamu hizmetleri yeterli düzeyde sağlanamamaktadır.

Günümüz yaşantısının gittikçe daha karmaşık olması bireylerin ve kurumların modern planlama, mühendislik tasarımı ve projelerin ışığı altında arazi, sualtı ve jeolojik yüzeylere ilişkin yoğun bilgilerle daha yakın olmasının, hızlı kesin ve ekonomik kullanımla aynı araştırmaları tekrar tekrar yapmaksızın alternatif çözümler üretmesini zorunlu kılmaktadır. Bu arayışların neticesinde de adı geçen ihtiyaçların karşılanması aracı olan "**Bilgi Sistemleri**" ortaya çıkmaktadır. Bilgi sistemleri, kullanım alanlarına ve amaçlarına göre, "Arazi Bilgi Sistemi", "Coğrafi Bilgi Sistemi (CBS)", "Kent Bilgi Sistemi" vb. gibi bölümlere ayrılabilir [1].

### **1.2 CBS Tanımı**

Bilgi elde etmek için, verileri daha önceden belirlenmiş biçimlerde anlık yöntemler ile kullanmak üzere saklayan sistemler "Bilgi Sistemleri" olarak adlandırılır. Bilgi sistemleri oldukça geniş bir kavramdır. Zamanla kullanım

alanlarına göre farklı bilgi sistemleri ortaya çıkmıştır. CBS (Coğrafi Bilgi Sistemi) de bunlardan birisidir. İdarelerin planlama, program yapma ve karar verme yetenekleri mekana ve mekana ilişkin bilgileri saklama yönetebilme imkanlarına bağlıdır.

Mekana ilişkin bilgilerin saklanması ve yönetilebilmesi de bir çok teknik disiplinde büyük başarılarla uygulanan bilgisayar yeteneklerinden birisidir. CBS adıyla anabileceğimiz bu yetenek, bilgini haritalar üzerinde saklanması, güncelleştirilmesi ve işlenmesi kapsamında pek çok uzmanlık alanında başarıyla uygulanabilmektedir.

Her bir kullanıcıya ve uygulama alanına göre değişen çok sayıda “Coğrafi Bilgi Sistemi (CBS)” tanımlaması mevcuttur. Bu sistemlere ait kayıtlar mekan referanslıdır. Daha doğru tanımlamalar ise fonksiyonel yetenekleri (CBS’de veri elde edilir ve işleme tabii tutulur) veya uygulamayı desteklemeleri ile ifade edilebilir.

Ancak CBS yalnızca bilgilerin yüklendiği haritaların bilgisayar ekranında hızlı ve çabuk şekilde görülebileceği bir sistem olmayıp, daha çok yoğun veri tabanı yönetimi bu veri tabanının, grafik veri tabanı ile ilişkilendirilmesi ve mekansal verilerin yönetilmesi desteklemek olarak ifade edilebilir.

Örneğin çizim ve benzeri gibi geometrik bilgiler, envanter, ruhsat ve benzeri öznelik bilgileriyle son derece başarılı olarak ilişkilendirilebilmektedir. Şehir planlama kapsamında, inşaat ruhsatları, tapu kayıtları, imar çap ve benzeri bilgiler imar planı üzerindeki herhangi bir parsel, binayla, hatta konutla ilişkilendirilebilmektedir [ 1, 2 ].

### **1.2.1. CBS kullanım alanları**

CBS uygulamalarına pek çok özel alanda gereksinim duyulur. Bunlara örnek olarak şehir planlama, demografi, kartografya, tarım, yangın ve salgın hastalıkların kontrolleri verilebilir. Her çeşit uygulama farklı özellikler, ölçekler ve değişken mekansal özellikler ile ilişkilidir. CBS uygulamalarını 3 ana gruba ayırabiliriz

- Sosyo-ekonomik uygulamalar: İnsanlar, altyapı ve araziye ilişkin mekansal analizleri toplayıp kapsar.
- Çevre uygulamaları
- Diğer iki çeşit uygulamanın yönetilmesi [ 2 ].

Yukarıda genel kullanım alanları verilen CBS' de kişilerin gerekli her türlü ihtiyacı karşılayabilecek kentsel ve kırsal çalışmayı destekleyen bir çalışmanın da bazı özelliklere sahip olması gerekir. Bu aşamada planlama amaçlı CBS' lerinden bahsedilir. Planlama amaçlı CBS de bulunması gereken bir sistemin;

- Gerekli tüm bilgileri eksiksiz içeren,
- Şeffaflık ilkesi çerçevesinde veri olarak ilgili düzenlemenin yasal ve teknik yürütülüş bileşenlerini kapsayan,
- Kolay anlaşılır,
- Verileri güvenilir,
- Verilerine kolay erişilir ve ekonomik ulaşılabilen yapıda olmasının gereği açıktır [ 3 ].

Toprağa ilişkin çalışmaların yürütülmesinde, toprak sahibi, yöneticiler ve politikacılar da planlamaya katılmalıdır. Denge çıkarları korunmalı, kamu hizmetinin de hızlandırılması sağlanmalı, ayrıca ne gibi bir çalışma yapılacak ise ona göre bir düzenleme yapılmalıdır. Ulaşılmak istenilen bilgi ve daha sonra kullanılacak veriler için gerekli düzenlemelerin de yapılması gereklidir. Daha sonra hukuki bir soruna yol açmaması için bir planlama sisteminde nelerin bulunması gerektiği en ince ayrıntılarına kadar düşünülmelidir.

### **1.3 CBS' de Kullanılan Veri**

Bilgi sisteminin temelini oluşturan veriler, bilgiye dönüştürülebilen her türlü işaret ile ifade edilebilen kazanımlardır. Veri, bir gözlem veya işlem sonucunda ya da bunlara karşılık olarak çıkmaktadır. Açık ölçmeleri, nokta numaraları, parsel sınırları, malik verileri gibi...

Bilgi sistemlerinde üretilen bilgiler, kullanıcı tarafından anlaşılabilir formlara dönüştürülmüş verilerden oluşan bir grup olarak tanımlanmaktadır. Bazı durumlarda bilgi, diğer bir bilginin elde edilmesiyle veri olarak kullanılmaktadır. Veri ve bilgi iç içe geçen iki kavramdır. Poligon hesabı sonucunda elde edilen koordinatlar, bilgi dir. Alan hesabında ise koordinatlar birer veri, hesaplanan alan ise bilgi olarak girilmelidir.

Haritalar yüzyıllar boyunca mekansal verinin görselleştirilmesinde kullanılmaktadır. Özellikle 1980'li yıllarda yazılımsal alandaki gelişmeler coğrafi bilgi sistemlerine de yansımış ve böylece mekansal verinin sorgulanması ve analiz edilmesi mümkün hale gelmiştir. Coğrafi bilgi sistemleri diğer bilgi sistemlerinden kullanılmakta olduğu veri yapısı bakımından farklılık gösterir. Bu sistemlerde kullanılan veri mekandaki konumları ile birlikte tutulur, diğer bir deyişle mekansal veri mekansal bir adrese sahiptir [ 4].

CBS'leri, mekan referanslı veriler için veri yönetimi ve erişim çalışmaları için anahtar sağlamaktadırlar. Mekan referanslı veriler zaman ve uzay özelliğini de taşır. Farklı formlardan, farklı kaynaklardan, değişik yerlerden gelen ve farklı yöntemlerle elde edilen verilerin CBS içinde entegre edilebilmesi için gerekli destek sağlanmalıdır [ 2, 5].

### **1.3.1. Mekansal verinin genel özellikleri**

Mekansal verinin temel özelliklerini inceleyecek olursak:

1. Mekansal veri, mekandaki planimetriye bağlı olarak nokta, çizgi, poligon ve bunların kombinasyonlarından oluşan karmaşık bir yapıya sahiptir. Mekansal verilerin ilişkisel veri tabanlarında iyi saklanamamasının nedenlerinden birisi de mekansal verinin sahip olduğu bu karmaşık yapıdır. Ayrıca mekansal verilerdeki operasyonlar standart bir ilişkisel operasyondan daha farklıdır.
2. Mekansal veri genellikle dinamikdir. Nesnelere dinamik olarak girilebilir, silinebilir veya güncelleştirilebilir. Bu yüzden veri yapılarının bu özellikleri etkili bir biçimde desteklenmeleri gerekmektedir.
3. Mekansal veri oldukça geniş bir alanı kaplamaktadır. CBS'ler günümüzde bir trilyon byte (terabyte) lık veri işleyebilirler.
4. Mekansal veriler için herhangi bir standart hesaplama yöntemi bulunmamaktadır. Bu durum herhangi bir standart operasyon kurulmasını engellemektedir. Geleneksel operasyonların büyük bir çoğunluğu özel uygulamalara dayanmaktadır. Buna rağmen günümüzde birtakım genel operatörler de kullanılmaktadır. Bu operatörlere örnek olarak kesişme (intersection) verilebilir [6].

### **1.3.2 Mekanın modellenmesinde kullanılan veri kavramı**

#### **1.3.2.1 Konumsal veri**

Konumsal veya geometri verisi, doğal olayın veya objenin geometrisini karakterize eder. Geometrik veri bilgisi koordinatlar, topoloji (A, B ve C'nin arasındadır) gibi bilgilerdir. Objenin sunumu ise noktasal, çizgisel, alansal ve hacimsel olarak yapılır.

#### **1.3.2.2 Öznitelik verisi**

Öznitelik verisi duyu organları ile algılanabilen (örneğin yaprak döken ağaçlar) veya başka aletler kullanılarak değeri veya varlığı belirlenebilen (örneğin sıcaklık) gibi özellikleri sunmaktadır. Öznitelik bilgileri nicel ve nitelik bilgileri olarak ayrılabilir.

#### **1.3.2.3 Zaman**

Mekansal veri zamana bağlıdır. Objeye ait öznitelik bilgisi zamanla değişebilir (örneğin bir yerleşim yerinin nüfusunun azalıp artması). Aynı zamanda olayın konumu da değişebilir. Planlama yapılırken zaman faktörü de göz önünde bulundurulmalıdır.

#### **1.3.2.4 Meta data**

Verinin verisi olarak ifade edebileceğimiz “*meta data*” verinin toplama yöntemi, sözel verinin kaynağı, tarih gibi veriye ait önemli bilgileri içerir [7,8].

### **1.4 CBS'nin Fonksiyonel Bileşenleri**

#### **1.4.1 Veri aktarımı**

Bu bileşen veri toplama, doğrulama ve kalite kontrol gibi faaliyetleri içerir. Yapılan ölçmeler, fotogrametri, uzaktan algılama çalışmaları, sayısallaştırma işlemleri vb. çalışmalar veri aktarımı ile ilgili konulardır.



### **1.4.2 Veri depolama**

Bu fonksiyon veri yönetim sistemleri (yedekleme vb.) dışında veri yapılandırma üzerinde yoğunlaşır.

### **1.4.3 Veri işleme**

- Semantik verilerin yani obje tanımlayıcı ve öznelik verilerinin veri tabanına girilmesi, değiştirilmesi, silinmesi,
- Nokta, çizgi tipi geometrik elemanların girilmesi, konumunun değiştirilmesi, ölçeklendirilmesi, döndürülmesi, simetrik transformasyonu, biçiminin değiştirilmesi, nokta sayısının azaltılması (weeding), bölünmesi (split), kopyalanması, birleşmesi (fusion) gibi bilgisayar destekli tasarımları,
- Poliline ve spline fonksiyonları
- Nokta, alan, çizgi detay bileşenlerinin (düğüm ve kenarların) topolojik derecelerinin değişmesi
- Benzeşim afin ve projektif dönüşümler
- Projeksiyon dönüşümleri
- Sınıflandırma, kodlama, tutarlılık ve format kontrolleri

### **1.4.4 Mekansal Analiz**

Çizgi ve yüzey entepolasyonları, yüzey modellendirme, istatistiksel analiz, zonlama (buffering), üst üste çakıştırma (overlay), yeniden sınıflandırma, alan birleştirme (disolve), ikili cebir (boolean), ağ analizi gibi işlemlerdir.

### **1.4.5 Sorgulama**

Coğrafi analiz, veri işleme ve veri sunma kapsamında sorgulama bileşeni görülmektedir.

### **1.4.6 Veri Sunma**

Toplanan ve CBS için üretilen verilerin raporlar, grafikler, haritalar, tablolar vb. şekillerde kullanıcıya sunulmasını kapsar [9].

## 1.5 CBS' nin Yararları

CBS' nin kullanımında kırsal ve kentsel uygulamalardan söz edilebilir. Özellikle kentsel çalışmalarda gelişmelerin getirdiği bazı karmaşıklıkların su-kanalizasyon, doğal gaz, yapı ruhsatı işlemleri gibi, kısaca belediyenin görev alanı içerisine giren ve yerel halkın sağlık ve sosyal yardımlaşmalarına kadar uzanan geniş bir yelpaze içerisinde düzenli bir sistemle desteklenmesini gereği açıktır. CBS, kentin arazi ve toprak bilgilerini, diğer teknik ve sosyal bilgilerini en sağlıklı ve en hızlı bir şekilde kullanıcıya ulaştırır. Sağlıklı bir kent planlaması, yönetimi ve gelişimi bu sayede gerçekleştirilebilir. Yapılabilecek çalışmalarda CBS' nin yararları aşağıdaki gibi belirtilebilir; Alt ve üst yapı şebekelerinin yapı onarım bakımı ve yenilenmesi gibi faaliyetlerde zaman ve maliyet tasarrufları sağlanabilir.

Gecekondulaşma, çarpık ve kaçak yapılaşma önenebilir. Belediye ve hazineye ait taşınmazların durumları ortaya çıkar. Bunların kullanım şekilleri, analizleri, kira bedelleri, taksit ve kontrolü sağlanmış olur. Arazi değerlerine ait bilgiler her an sorgulanabilir. Kentin ulaşım itfaiye hizmetleri ile doğal gaz şebekelerinin kontrolü gibi hayati önemdeki hususlar denetim altına alınmış olur. Ayrıca taşınmaz mal alımı ve satımlarında ortaya çıkan çarpıklıklar ortadan kalkabilir ve bu işlemin sunucu olarak vergi kayıpları önlenir [ 10 ]. Özellikle iyi yapılandırılmış ve kurullar arası ilişki sağlanmış, sağlam temelli bir CBS oluşturmak pek çok sorunların hızlı, herhangi bir hukuki şüpheye yer vermeyecek şekilde anında çözülmesini sağlayacaktır. Planlamalar daha düzgün yapılabilecek, zaman ve maliyet kayıpları en aza inecektir. CBS sayesinde veri paylaşımı yapılabilecek, etkileşimli çalışma sonuçları ortaya çıkacak, karar verme aracı olarak CBS kullanılacaktır. Ayrıca denetimin sağlanması sonucunda dolaylı yoldan da olsa gelir artışı sağlanabilecektir.

## 1.6 Türkiye' de CBS Ve Uygulamaları

CBS' nin Türkiye' deki mevcut gelişimi 1990'lı yılların özellikle ortalarından itibaren hız kazanmaktadır. CBS üzerine yatırım yapan kuruluşlar 3 ana grupta sınıflandırılabilir;

- Merkezi idareler
- Mahalli idareler

- Özel sektör

Her üç grupta gerek işledikleri veri tipleri, gerek CBS’den beklentileri ve gerekse insan kaynakları açısından farklılıklar göstermektedir. Bunlara ait birtakım ayrıntılar aşağıda verilmiştir.

### 1.6.1 Merkezi idareler

Daha çok küçük ölçeklerde çalışmaktadır. Oluşturulan veri tabanlarında “Sürekli Veri Tabanı” ihtiyacını, verilerin birleştirilerek sürekli kullanılması yaklaşımı ile karşılamaktadır. Bu grup, insan kaynakları açısından daha şanslı görülmektedir. Konuya yakın insanları, mesleki kaynakları ve alt yapısı itibarıyla daha iyi koşullarda çalışma ortamları oluşturabilmektedir. Bu grupta yer alan kurum ve kuruluşlarının he men he men tam mında, en az bir birinde kuşatılmış olan yazılı mve donanı malyapısı mevcuttur.

### 1.6.2 Mahalli idareler

Daha büyük ölçekli i mar planı, halihazır harita, kadastr o başta olmak üzere zaman zaman altyapı ve bunlara ilaveten emlak, çevre temizlik, ruhsat gibi sözel bilgilerin erişimine yönelik olarak CBS’ye eğilim göstermektedirler. Bu grupta CBS’den temel beklenti, i mar, harita, planlama otomasyonunun sağlanmasıdır.

### 1.6.3 Özel sektör

Özel sektör yapısı gereği, ya yatırımı yaptığı konudan doğrudan gelir elde etme veya bu yatırımı araç olarak kullanarak elde edeceği faydayı rantta dönüştürebilme durumunda yatırımı yönelmektedir. Bu hususta CBS’nin nekana yönelik analiz araçlarının kullanımı öne çıkmaktadır. Ancak, bu noktada CBS’ye yapılacak olan yatırımın kullanılacak verinin elde edilmesi için harcanacak bedelin çok altında kalması önemli bir problem olarak karşımıza çıkmaktadır.

Veri yatırımının büyüklüğü, veri eksikliği, veri hataları gibi temel sorunlar yüzünden özel sektör bu yatırımı malanına şu anda pek sıcak bakmamaktadır [ 1 ].

Türkiye’de Tapu- Kadastro Genel Müdürlüğü de bazı çalışmalar yapmaktadır bu çalışmalardan birisi de TAKBİS adı altında yürütülmeye çalışılmaktadır [ 10 ].

CBS sistemlerinin bazı başarısızlıklarında iyi araştırma yapılmadan alınan sistemlerin daha aktif kullanıma geçmeden kullanılmaz duruma gelmeleri yer almaktadır. Ayrıca kurumlar arası iletişimsizlik ve her kamu veya özel şirketlerin kendi CBSlerini oluşturmaya çalışmaları sonucunda verilerdeki tutarsızlık ta göze çarpmaktadır.

## 2. MODELLEME DİLLERİ

Veriyi tanımlama ve kullanma şeklini (kayıtların aranması için gerekli erişim yollarını gösteren ilişkileri) açıklayan kavramsal organizasyona “ Veri Modeli (Kavramsal Model)” denir [ 5].

Kavramsal modelleme ve analog haritaları karşılaştırmak mümkündür. Bilgisayarlar ve bilgisayar dilleri soyut bir doğaya sahiptirler. Bu yüzden yapılan bir programın genel modelini elde etmek için şekiller ve diyagramlar kullanılır. Kavramsal modelde diyagram(boxology) grameri kullanılmaktadır. Aynı şekilde haritalarda göz önünde bulundurabilir. Bu durumda kavramsal modeldeki diyagram gramerinin yerini haritalarda lejantlar ve işaretler almaktadır. Veri modelinde de aynı haritalarda olduğu gibi objenin veya olgunun önemli yanları soyutlanmaktadır.

Modelleme dilleri genel de yapısal prensiplerine veya dilin perspektifine göre 7 farklı sınıfa ayrılabilir.

### *Yapısal yaklaşım*

Veri ve veri modellemesi üzerinde odaklanmaktadır.

### *Nesneye dayalı yaklaşım*

Nesneye dayalı perspektif nesneye dayalı programlama dillerinin desteklenmesi ihtiyacından oluşmuştur. Bununla birlikte nesneye yönelik analiz ve tasarım başlı başına ayrı bir uzmanlık haline gelmiştir.

### *Davranışsal yaklaşım*

Bu perspektifin ana düşüncesi durumlar ve bir durumdan diğerine dönüşüm yani geçişlerdir. Bu perspektif ile yaşanan en büyük problem karmaşık sistemlerin yönetilememesidir. Bu problemin üstesinden gelmek için bazı diller AND ve XOR ayrışmalarına hiyerarşik soyutlama mekanizmaları eklemişlerdir.

### *Kural yaklaşımı*

Bu perspektifin ana uygulama alanları bilgi sistemleri ve yapay zekadır. Genelde kural If (koşul) then (sonuç) şekline sahiptir.

### *Fonksiyonel yaklaşım*

Veriden ziyade yöntemler üzerinde odaklıdır. En iyi bilinen kavramsal modellemeli, veri akış diyagramları yöntemidir.

Yukarıdakilerin haricinde iletişim ve aktör rolü perspektifleri bulunmaktadır.

Açıklanan yaklaşımların çoğunun bilgi sistemlerinin geliştirilmesinde gerekli olduğu açıktır. Davranış modelleri zaman geçtikçe objelerin nasıl değiştiğini ifade eder, fonksiyonel modeller bu işlemler esnasında kullanıcıyı destekler. Diğer taraftan yapısal modeller gerçek dünyanın statik davranışlarını sunarlar. Nesneye dayalı yaklaşım yapısal yaklaşımlar ile yakından ilişkilidir, çünkü bu model nesnelere arasındaki ilişkileri tanımlamaktadır. Kural yaklaşımları, değişken bilgi sistemleri için oldukça değerlidir.

## **2.1 Yapısal Modelleme**

CBS’de kullanılan “yapısal modelleme”de “nesneye dayalı modelleme”nin aksine obje “veri ve fonksiyon” olarak ifade edilir. Yapısal modellemenin odağı veri ve veri modellemesi, ana içeriği ise varlıklar, ilişkiler, öznitelikler ve ilişkilerdeki kısıtlamalardır (kardinallik).

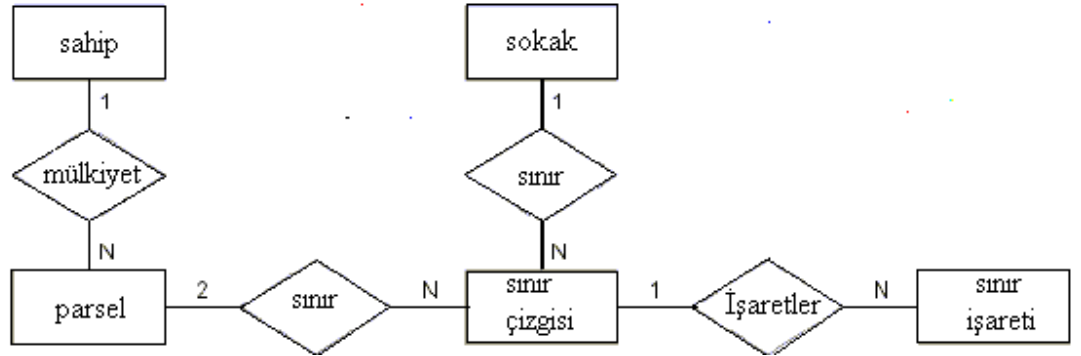
ER (Entity Relationship) Chen tarafından geliştirilmiştir [12]. Bilgisayar dünyasında fazla kullanılmasının nedeni, basit diyagramatik sunumu ve ilişkisel veri tabanı tablolarına kolay geçişinden dolayıdır. ER modeli zayıf veri tabanı yapısına rağmen gerçek dünyaya ait verilerin modellenmesinde oldukça iyidir. ER modelinin iki elemanı bulunmaktadır. Bunlar:

- a. İlişkiler
- b. Öznitelikler

Chen varlığı “farklı olarak tanımlanabilen şey” ve ilişkiyi de “varlıklar arası birlik” olarak tanımlamıştır. Her varlık öznelik kurulları ile karakterize edilmiştir. Bir ER diyagramı ( Şekil 2.2 ) ile bir kadastraya ait parsel, kişi ve sokaklar ( Şekil 2.1 ) arasındaki ilişkiler şekillerle tanımlanmaktadır.



Şekil 2.1. Parsel ve sokakların gösterildiği bir kadastr haritası



Şekil 2.2 Yapısal modelleme ile yapılmış bir kadastr işlemi [12]

Model, ( Şekil 2.1 ) deki kavramı dayanmaktadır. (Şekil 2.1) de sokaklar ve parseller gösterilmiştir. Bununla birlikte gerçek dünya ER dili ile modellenirse bazı eksikliklerle karşılaşılır. EER (Extended Entity Relationship) dili ise ER modelinin bir uzantısıdır alt tip ve birlik kavramlarını da içerir. Bu günümüzün nesneye dayalı yaklaşımın bir parçasıdır. Yapısal programlama dillerine örnek olarak FORTRAN, PASCAL, C, BASIC ve ADA verilebilir [ 3, 12 ].

## 2.2 Nesneye Dayalı Yaklaşımlar

Nesneye dayalı yaklaşım ilişkisel modele bir alternatif olarak bilgi sistemleri için bir paradigma olmuştur. Bu yaklaşım bilgisayar sistemlerine farklı seviyelerde uygulanabilir.

- Nesneye dayalı programlama dilleri
- Nesneye dayalı analiz ve tasarımlar
- Nesneye dayalı veri tabanları ve veri tabanı yönetim sistemleri olarak gruplandırılabilirler [ 13 ].

## 2.3 Nesneye Dayalı Programlamanın Temel Amacı

Yakın zamanda nesneye dayalı yöntemlerle çalışmalar bilgisayar ekranlarında yeni olanakların kapılarını açmıştır. Bu yeniliklerden CBS'lerde etkileneşlerdir. Eskiden geleneksel harita veri tabanının da pasif veri olarak tutulan koordinat ve öznitelik karakterleri günümüzde aktif olarak işleme görmektedir.

Nesne temelli programlama, program veya genel ortamda yazılım sistemlerinin geliştirilmesinde yeni bir yöntemdir. Nesne temelli programlamanın esas amaçları, yazılımın kolay büyüyebilmesini sağlamak, daha önceden yazılmış olan yazılımın tekrar kullanımını sağlamak, yazılımbakımından ise karşılıklığın ve maliyetin en aza indirilmesidir. Nesne temelli programlama kullanıldığında, yazılımın sistemin tasarımı aşamasına daha sıkıca bağlandığı görülmektedir. Nesne temelli programlama kavramının başarılı olmasının arkasında, temel düşünce tarzının insanların doğal arı gereği sahip oldukları düşünce tarzına olan benzerliği de yatmaktadır. Bugünkü anlayışa göre objelerin nitelikleri (geometrik ve özniteliysel) ve fonksiyonları biçiminde tek varlık olarak modellenmesi, insanların bunları belleklerinde konuşma dili yardımıyla kavramalarına benzer bir yapıyı yansıtmaktadır. İnsanlar çevrelerini nesnel topluluğu olarak algılar ve nesnelar arasındaki ilişkileri sorgularlar.

İnsanların problem çözmeye kullandıkları temel bir nesnedir. Bu nedenle eğer programı klasik programlama yöntemlerini bırakıp, nesne temelli bir platformda problemlerini çözmeye çalışırsa, programlılık açısından yeni olan bu düşünce tarzının gerçekte kendi düşünce tarzına çok benzer olduğunu görecektir. Böylece bir



CBS programı ile, bunu kullanan kişilerin gerçek doğa ile düşünüş modelleri arasındaki benzerlik sağlanmış olmaktadır.

Sonuçta ise, kullanıcı sistemle aktarılmak istenen verileri kolay alımlayabilmektedir. Bu avantajın en önemli dayanağı, daha önce belirtildiği gibi, ilgili somut veya soyut objenin yapısal ve fonksiyonel özelliklerinin bir bütün olmasıdır [ 3, 14, 15 ].

## 2.4 C++ Dili

C++ dili C dili üzerinde geliştirilmiştir ve C dilinin bir üst aşamasıdır. C++, C'nin gücünden, esnekliğinden ve etkinliğinden faydalanır. C++, C ile tam bir uyumluluk içindedir. C++ dili kullanılarak, C kodu yazılabilir ve geliştirilebilir. Ancak C++, C diline nesne temelli programlama desteği vermesi ile büyük değer kazanmaktadır.

C++, hibrit veya karma bir dil olarak nitelendirilebilir. Hem yapısal C dilinin özelliklerine ve hem de nesne temelli programlama yeteneğine sahiptir. Tipik bir C++ program, nesne temelli programlama içinde yapısal programlamayı da efektif olarak destekler.

C++ bazı yanlarını, BCOL ve Simula 67 dillerinden esinlenerek kazanmıştır. Örneğin alt sınıf veya türetilmiş sınıf kavramı Simula'dan alınmıştır. Dilde önceden tanımlı operatörlerin tekrar tanımlanması, yani operatör yinelenme olarak ta bilinen özellik ve uygulanmanın herhangi bir yerinde bir değişken ilk kez kullanıldığında tip bildiriminin yapılabilmesi özelliği Algol 68'den alınmıştır. Diğer modern diller gibi, kendisi dışındaki programlama dillerinin en iyi özelliklerini almıştır. Yine de en yakındığı dil C'dir [ 15 ].

## 2.5 Nesneye Dayalı Yaklaşımın Temelleri

“Nesne”, nesneye dayalı yaklaşımın merkezini teşkil etmektedir. Nesneye dayalı sistemlerde ilişkisel modellerde olduğu gibi sistemin statik davranışı olduğu kadar, ilişkisel modellere ek olarak dinamik davranışlar bulunmaktadır. Nesneye ait statik özellikler öznitelikler ile sunulur. Öznitelikler daha önceden tanımlanmış alanlardan değerler alabilir. Bir şehir objesi isim merkez, nüfus gibi özniteliklere sahip olabilir. Örnek olarak “Londra” “İsmi” özniteliği için değer olarak verilebilir. Öznitelikler

objenin durumunu sunarlar. Objeye ait dînamik yön, yani davranışlar ise operasyon takımları tarafından sunulur. Operasyonlara karşılık olarak bir obje uygun şartlar altında performans gösterecektir. Örneğin çizimi yapılacak olan bir bölgenin ekran üzerinde sadece gösterimini yapması gibi bir fonksiyona yeti nilmez aynı zamanda farklı ölçeklerde ilgili bölgenin çiziminin yapılması, genelleştirme gibi farklı operasyonlar da uygulanabilmektedir.

Böyle operasyonlar bölgenin alanının ve çevresinin hesaplanması, farklı ölçeklerde bölgeye ait çizimlerin yapılması ve farklı seviyelerde genelleştirmeyi, sistemden bölgele rin oluşturulması ve silinmesini de içerir. Nesneye dayalı yaklaşımlar için anahtar düşünce aşağıda ifade edilmiştir [13].

$$\text{Obj e}=\text{dur u m}+\text{fonksi yonluk} \quad (2.1)$$

## 2.6 Nesneye Dayalı Programlamada Kullanılan Nesne

Nesne, bir varlığın sunumunda özel olarak tanımlanır (obje kimliği), güncelleştirilebilir bir duruma sahiptir (obje özniteliği) ve davranışları tarafından karakterize edilirler (obje davranışı). Buna ek olarak, bir nesne diğer nesnelere gibi aynı kimliğe değil, aynı yapısal ve davranışsal karakterlere (sınıflandırma) ve diğer objeler ile ilişkilere sahiptir.

Yukarıda sayılan obje özellikleri gruplandırılacak olursa, obje kavramının hesaplamadünyası içinde nasıl gösterilebileceği anlaşılabilir.

- Tamıncı özellikler: Öznitellik bilgileri objenin tarif edilmesinde kullanılır.
- Prosedür özellikleri: Davranışların operasyonlar ile sunulmasıdır.
- Yapısal özellikler: Bu konuya ilişkin kavramlar sınıflar, miras, ilişkiler ve toplamalar olarak karşımıza çıkar.

Nesne tanımlanabilir ve operasyon talebine göre davranış gösterir ve nesneye dayalı sistemlerde her gerçek dünya varlığı aynı tarzda obje tarafından sunulur. Obj e özel olarak kimliklenebilen, öznitellikleri güncellenebilen, aynı zamanda davranışları tarafından karakterize edilebilen, varlığın sunumu olarak anlaşılır. Bütün bunlara ek olarak bir obje aynı yapısal ve davranışsal karakterlere (sınıflandırma) ve diğer nesnelere sahiptir.

Obj e, belli bir s ınfa ait olan bir deęişken olarak ele alınabilir. Obj e, kendi s ınıfında tanı m yapılan soyut veri tipi nin bir örneğidir. S ınıf tanı mında var olan veri yapıları ve yöntemler, özel ve dışa açık olanlar, bu s ınfa ait olan objeye de geçmektedirler. O halde her obj e, ait ol duğu s ınıfın büt ün karakterini üzerinde taşı maktadır. Obj e, s ınıf tanı m i çinde tanı mlanmış olan (özel ya da dışa açık) büt ün verilerin birer kopyasına kendisi de sahiptir. Bunlar, nesnenin belli bir andaki durumuna sahiptirler ve o an için ne durumda ol duğunu gösterirler [ 15, 16 ].

## 2.7. Nesneye Dayalı Veri Modellemesinin Özellikleri

Yeni nesneye dayalı dünya da statik özellikler yerini dinamik özellikler ile değiştirmiştir. Uygulama, çizilmiş özellikler yerine özellik kodu ve saklanmış koordinatlardan elde edilen düzenlenmiş sunuma göre yapılır. Uygulamada her objeye mesaj yollar ve örneğin objeden kendisini çizdirmesi istenir. Her obje, kendine ait büt ün bilgiyi kullanarak en iyi nasıl çizilebileceğine karar verir. Bu bilgiler öznitelikler ve diğer objeler ile olan ilişkileri de içerebilir. Bunun anlam ise tek bir coğrafi veri takımından, geniş bir üretilmyapılabileceğidir.

Nesneye dayalı veri modeli obje sınıf hiyerarsisini ( miras) ve nesne bindirme (operatör overloading) destekler.

Nesne kavramının nesneye dayalı veri modellemesinin merkezi ol duğu daha önceden de belirtilmişti. Bir obje herhangi bir öznitelik değerinden bağımsız olarak kimliğe, özel durumlara ve operasyon takımları olan genel ara yüzlere sahiptir. Objeler arasındaki etkileşimin tek anlam, mesaj geçişidir. Aynı davranışlara ve yapılara sahip olan objeler, sınıflar içinde organize edilirler.

Nesneye dayalı veri modeli ile topolojik ve mekansal özellikli nesnelere arasındaki ilişkiler daha basit bir şekilde tablolara ihtiyaç olmadan doğrudan yürütülebilir. Bu, nesneye dayalı yaklaşımlara ait kimikleme, veri saklama ve soyutlama prensibi (encapsulation), miras ve çok biçimlilik (polimorfizm) özelliklerinden dolayıdır.

Yazılım mühendisliğinden bilinen nesneye dayalı modellemeyi karakterize eden özellikler aşağıda sınıflandırılmıştır.

Obj e Sınıfları (Soyut Objeler)

- Karşılaşık (Kompleks- Bir den Çok Obj e ni n O luşt ur du ğ u ) Obj e ler
- Obj e Kı mlı ğ ı
- Obj e Yapısı ve Belli Duru ml ar da ki Dav ranı ş lar ın Saklı Tut ul ma sı ya da Veri Sak la ma Ve Soyut la ma Prensi bi ( Encapsul ati on)
- Ot omatik Ö zniteli k Ve Fonksi yon Ü st len me ( MİR as)
- Çok B i ç i mlilik (Poli morfozl uk) [ 3, 12, 13, 17 ].

### 2.7.1 Obj e S nıfları

Yeni varlıkları (objeler) di namik olarak oluşt urmak için araçlara ihtiyaç duyulur. Objeler, durumlar (öznitelikler) ve davranışlar (operasyonlar) tarafından tanımlanır. Bunlara ek olarak, aynı davranış ve özniteliklere sahip olan objeler gruplandırılır. Ayrıca bir varlığın diğerlerinden ayrılmasını sağlayabilmek için, genelleştirme ve özelleştirme mekanizması kullanılır. Eğer bir sistem yukarıda sıralanan özellikleri içeriyor ise soyut yaklaşım var demektir.

Karşılaşık problemleri anlamak ve çözmek için soyutlama kullanılır. Bilgisayar sistemleri normalde bu konularda insanları desteklemek için kullanılırlar, fakat uygulama ve yöntemler farklıdır. Bugün pek çok programlama dili, yeni veri tipleri oluşturabilmeyi desteklemektedirler. Veri soyutluğu, tip kavramının genişletilmesidir. Böylece soyut operasyonlara ve soyut veri nesnelерinin tanımlanmasına izin verilmektedir.

Zorunlu sistemler (yapısal modeller) (ALGOL ve PASCAL gibi) iki farklı bileşene dayanarak problemlerin çözümünü için çalışır. İlk olarak sahip olunan veri yapıları, modellenen varlıkların durumlarını (öznitelikler) sunar. İkinci olarak sahip olunan prosedürler ve fonksiyonlar bu yapılar ile çalışır. Fakat veri yapılarını ve bu yapılar içindeki obje ile çalışan prosedürleri gizlemek için bir yapı yoktur. Bu programların yanında Modula ve ADA gibi diller ise soyutluğu daha iyi destekler.

Nesneye dayalı programlamada her nesne; sınıf (class) adı verilen ve kullanıcı tarafından tanımlanan veri tipinin örneğidir. Sınıfın soyut veri tipi olarak tanımlanabilebilmesinin nedeni yukarıdaki ana özellik olan kullanıcının kendi sınıfını tanımlayabilmesi özelliğidir. Diğer programlarda ve özellikle nesneye yönelik programlama dillerinde veri soyutlamanın temel amacı “bilgi saklamaktır”. Birbirleri ile ilişkili fonksiyonlar (nesneye dayalı programlamada yöntemler veya

operasyonlar olarak ifade edilmiştir) veya işlemler, veriler ile birlikte bir sınıf yapısına yerleştirilir. Nesnenin tüm özellikleri ve davranışları tanımlı olduğu sınıf tarafından belirlenir.

Nesnedeki veriye erişim fonksiyonları ve işlemleri içeren bir arayüz yardımıyla yapılır. Arayüz genellikle nesnenin ait olduğu soyut veri tipinde yani sınıfta tanımlanır. Bu şekilde tanımlanmış olan veri soyutlama işlemi, veriyi uygunsuz işlemlerden korur.

Kısaca, sınıf tanımlama soyut veri tipinin iç veri yapısını ve bu veri tipinin üzerinde kullanılacak yöntemleri topluca tanımlar. Objeye sınıfı, uygulama yapısı ile ilgili iken, obje tipi semantik düşünce ile ilgilidir. Aşağıdaki yapılar objelere ait özellik ve uygulama işlemlerini vermektedir.

<u>Tanımlama</u>	<u>Uygulama</u>
Obje tipi	Obje sınıfları
Operasyonlar	Yöntemler
İlişkiler	Mesajlar

Obje sınıfları altında, ilişkiler ya da objelerin konuyla ilgili önemli özellikleri altında birleştirilmesi kastedilmektedir. Örneğin tüm parsellerin alansal büyüklük, biçim ve maliyetlerden bağımsız olarak bir obje sınıfı altında toplanması gibi. Objenin mutlaka somut karakterde ya da CBS den genellikle alışık olunan biçimde mekansal olması gerekmemektedir. Örneğin bir taşınmazın satış işlemleri (satış işleminin yapıldığı müdürlük, satış tarihi, eski malik, yeni malik ve yevmiye numarası gibi bileşenlerden oluşan), obje gibi ele alınabilecek bir olaydır. Yine kadastraya amaçlı bir CBS de “malikler” obje sınıfının olabilmesi gibi.

Bütün objeler obje sınıflarına bağımlıdır. Her sınıf için birden fazla obje bulunabilir, fakat her obje sadece bir sınıf içindedir. Objeye tarafından ne gibi değerlerin tutulabileceğini, sınıf tanımlar. Değerler basit veri tipleri (tam sayılar, karakterler ve tarihler gibi.) olabileceği gibi, daha özel tiplerde (geometrilere, yerler, raster veriler ve tablolar) olabilir. Objeye bütün bunlara ek olarak objeler arasındaki yapısal bilgiyi veya kaynakları tutabilir.

Nesneye dayalı veri programlaması soyut veri tipleri ve mirasın birleşimidir.

OOP=Object Oriented Programming (nesneye dayalı programlama)

ADT=Abstract Data Type (Soyut veri tipi)

Sınıf bir şablon tarzıdır. Bir sınıflama mantığı söz konusudur. Programda kullanılacak belli nesnelere, ortak bir tarzdan türetilmeye çalışılır. Bu sistemin gerçek hayata uygulanması da aynı şekildedir. Örnek olarak adres oluşturulmasına yönelik bir sınıf örneği verilebilir. Nesneye dayalı modellerde sadece genel olarak bir objektin yapısı yapılır. Daha sonra gerekli detaylar eklenir. Aşağıda anlatılanların nasıl gösterileceğine dair bir yapı bulunmaktadır. Aşağıda C++ programlama dilinde “Adres” sınıfının oluşturulmasına ait bir örnek bulunmaktadır.

```
#include <iostream>
#include <string>
using namespace std;
class adres {
    char isim[40];
    char sokak[40];
    char posta kodu[10];
public:
    void sakla(char *i, char *s, char *p);
    void göster();
};
void adres::sakla(char *i, char *s, char *p)
{
    strcpy(isim, i);
    strcpy(sokak, s);
    strcpy(posta kodu, p);
}
void adres::goster()
{
    cout << isim << "\n";
    cout << sokak << "\n";
```

```

        cout << postakodu << "\n\n";
    }
int main ()
{
    adres a;
    a.sakla("Ahmet", "Ççek", "80000");
    a.göster();
return 0;
}

```

[ 3, 14-16, 18, 19 ].

### 2.7.1.1 Obje sınıflarında kullanılan operasyonlar

Bir objenin sadece yapısal nitelikleri önemli değildir. Örneğin bir obje ile insanların hangi işlemleri yapabileceği ya da diğer bir ifade ile, bir obje için uygulanabilecek operasyonlar da objelerin parçaları olarak ele alınır.

Operasyon olarak, bir parselin emlak vergisi değerini hesaplamak için kullanılan "fonksiyon" (değer oluşturan fonksiyon) ve "prosedür" olarak, inşaat ruhsatı istemini kabulü ya da reddi işlemleri için gerekli adımlar düşünülebilir.

Obje üzerindeki çalışmalar, objenin sınıf tanımında tanımlı olan yöntemlerden biri veya birkaçı uyarılarak gerçekleştirilir. Nesne temelli programlamada bir yöntemi uyararak, yönteme mesaj yollamak olarakta bilinir. Dışarıdan sadece objeye mesaj göndererek objenin nasıl davranması gerektiği belirtilir. Nasıl yöntemler, nesne temelli olmayan dillerdeki prosedür ve fonksiyonlara benzetiliyorsa, yöntemle yollanan mesaj da, prosedür veya fonksiyonlar çağrılırken kullanılan parametreler gibi parametreler içerir. Bir yöntemin uyarılması veya yöntemle mesaj gönderilmesi, genellikle objede saklanan veri üzerinde değişikliği oluşturur. Bu nedenle objenin yapısı içindeki veriler, objenin oluşturulduğu andan itibaren geçirdiği yaşantının bir anlamda sonucudur. Burada objenin yaşantısından kastedilen, çeşitli parametrelerle objenin yöntemlerine uygulanan mesajlar ve neden oldukları değişikliklerdir.

Farklı sınıflara ait objeler, kendilerine gönderilen mesaj aynı olsa bile bu mesaja olan tepkileri farklı olabilir. Objeye mesaja kendi yönteminin gerektirdiği biçimde yanıt

verecektir. Bu da yine insan düşünce mantığına yakındır. İnsanın gerçekleştirmek istediği işlem kullanılan ifadeler farklı olsa da aynı amaca yönelikse, farklı objeler için bu işlemin farklı sonuçlar vermesi insan düşüncesi ne uygun değildir.

Bir objenin davranışları o objenin operasyonları tarafından tanımlanır. Bir yöntem kesin objeler için operasyonun uygulanmasını desteklemektedir. Ayrıca farklı objeler, aynı operasyon için farklı yöntemlere sahip olabilir. Bu mekanizma “yöntemlerin farklılaştırılması (overriding)” olarak adlandırılmaktadır. Bir objeye bir mesaj gönderildiği zaman bu obje yöntemi çalıştırır ve diğer bir objeye mesaj yollayabilir. Mesajlar obje etkileşimi için kullanılırken, yöntemler bir mesajdaki reaksiyon olarak çalışırlar. Yöntemler, objeler arasındaki etkileşimi için sadece arayüzdür. Bazen objenin özneliği sistemin gizlenmesini (encapsulation) desteklemesi doğrultusunda doğrudan işlenebilir. Bazı programlar ise, yöntemler öznelik değerleri tarafından doğrudan desteklenmelidir. Bu sistemler güçlü gizleme (encapsulation) özellikleri gösterirler.

Yöntemler farklı özelliklere sahiptir. Her bir yöntemine de adı geçen özelliklerin birleşiminden oluşmuştur. Özelliklere ait kısaca bilgi vermek gerekirse. Bunlar:

- Geri dönen değerler: Geri dönen değerlerde yöntemler sadece bilgini geri alımlında kullanılır.
- Yan etkiler: Objeye ait öznelik işlenmesinde kullanılır.
- Parametreler: Bazı yöntemlerde parametre bulunur, bazıları da ise bulunmaz. Eğer bir yöntem parametreye sahip ise, yöntem ön hesaplanabilir bir özelliğe sahiptir. Parametresiz bir yöntemi için ise sadece bir sonuç ortaya çıkmaktadır.

### **Çoklu yöntemler**

Çoklu yöntemler çoklu polimorfik fonksiyonlar olarak anlaşılır. Çoklu yöntemler, kullanıcıya aynı isimle, fakat farklı parametreler ile farklı yöntemleri uygulayabilme imkânı sağlarlar. C++ da çoklu yöntemi için “+” veya “-” operatörleri örnek olarak verilebilir. Bazı sistemler sistemin operasyonu süresince kullanıcıya yöntemleri değiştirme izni verirken, C++ gibi bazı diller bu özelliği desteklemezler [ 3, 15, 16, 20 ].



## 2.7.2 Objeler arasındaki ilişkilerin kurulması

Objelerin birleşimi gerçek dünya yapısında ve sistemde oldukça önemlidir. Objelerin karışık iç yapısı ile modellenmesine izin verir. Nesnelerin toplanıp bir araya getirilmesinde çeşitli yollar vardır. Bunlar;

- Agregasyon
- Asosiyasyon
- Düzenli birleştirilmiş nesnelere

### 2.7.2.1 Agregasyon

Bütün bir obje tek bir obje olan "taşınmaz mal", "parsel" ve "oturlardan" oluşur. Objeye eğer agregasyon işlemi uygulanırsa bir obje, semantik olarak genişletilmiş duruma gelir. Bazen ilave kısıtlamalar zorla eklenebilir.

Agregasyon ilişkisi, karışık objelerin oluşturulmasında kullanılır. Bu işlemde objeler parçalardan oluşmuştur. Bu parçalar ise kendilerine ait parçalardan oluşur. Bir belgenin paragraflardan, paragrafların satırlardan ve satırların karakterlerden oluşması karışık objelere ait bir örnek olarak verilebilir. Objeler ve alt objeleri arasındaki ilişkiler, alt objelerin prosedürler tarafından tanımlandığı prosedür sunumları, bir objenin alt objesinin listelerini sakladığı obje kimliği sunumu ve alt objelerin doğrudan obje içinde saklandığı değer tabanlı sunumlar olarak ayrılabilir.

Bu aşamada aynı zamanda karışık objeler, bağımlı objelerden söz etmemek mümkündür. Karışık objeler, karışık objelerin hiyerarşisi ile bir obje olarak görev yapar. Bir araç için bu örneği vermek uygun olur. Aracın gövdesi ve kapılar ayrı ayrı aracın birer parçasıdır. Bunun ışığında arabamın karışık obje olduğunu söylenebilir ve karışık obje, obje hiyerarşisinin köküdür. Bağımlı objeler ise, diğer objelerden oluşmuştur. Belge, bu durumu için güzel bir örnek olarak verilebilir. Buna göre belgeyi oluşturan paragraflar sadece belge olduğunda mevcuttur. Belge silinirse, paragraflar da silinir.

## 2.7.2.2 Asozasyon ilişkisi

Bölge tipi objesi ile farklı bölge objelerinin ilişkileri olabilir. Mekansal objelerin aralarındaki ilişkiler incelenecek olursa nokta-nokta ilişkisi olarak komşusudur, dağıtılmıştır nokta-çizgi ilişkileri için yakınında, nokta-alan ilişkisi için merkezinde, içinde çizgi-çizgi ilişkisi geçer, birleşir, çizgi-alan ilişkisi kesişir, sınırında ve alan-alan ilişkileri için çakışır gibi ilişkiler mevcuttur. Bunun yanı sıra sağında, solunda, yanında, üstünde, aşağısında, arkasında, önünde, yakınında, uzakta, arasında gibi çeşitli ilişkiler de ifade edilmektedir. Nesnelere arasındaki ilişkilere biraz açıklık getirmek gerekebilir. Eğer bir obje mantıki olarak bir veya birden fazla nesne ile ilgili ise objeler arasında ilişki den söz edebilir.

- İlk ilişki objenin özniteliklerine uygulanabilir. Örnek olarak bir işyeri verilebilir. Her kişi işveren özniteliğine sahip olabilir. Diğer taraftan her şirket, bu işyeri için çalışan kişilerden kurulu bir işçi özniteliğine sahiptir. Bazı sistemler bu ilişkileri tersine durumlarda da destekler. Bu oldukça önemli bir özelliktir. Çünkü bir öznitelik güncellendiğinde diğeri de otomatik olarak güncellecektir. Böylece sistemin güven vermesi sağlanır.
- İlişki gösterimine ait diğer bir olasılık ise ilişkinin obje gibi uygulanmasıdır. Öğrenci, üniversite ve ders arasındaki ilişkiler için ikiden daha yüksek dereceli ilişkilerden söz edilebilir.

Bazı mümkün olabilecek ilişkiler incelense:

### a. İsteğe bağlı seçim ( *Optionality* ):

Böyle bir ilişki sayesinde isteğe bağlı ve zorunlu ilişkiler birbirinden ayrılabilir. Örnek olarak kişi ve işveren örneği verilecek olursa bütün kişilerin bir işveren olması gibi.

### b. Kardinallik:

İlişkiler bazı seviyelerde gösterilebilir. 1:1, 1: m m n

### c. Transfer edilebilirlik:

İlişkiler transfer edilebilir ve edilemez olarak ayrılabilir. Buna göre kişi ve şirket arasındaki ilişki verilebilir. Eğer transfer edilemez notu eklenirse kişi işvereni değiştiremez.

### d. Semantiklik

Sistemler ilişkiler arasındaki semantikliği destekleyebilirler [13, 16].

### 2.7.3 Obj e K i m i ğ i

Obj e k i m i ğ i onu di ğ erlerinden ayıran özelli ğ i dir ve bir sistemde nesne k i m i ğ i ile var olur ve bu var olu Ŗ, de ğ erlerinden ba ğ ı nsı zdır. Nesne k i m i ğ i, ana bir sistem yapısıdır. Buna ra ğ men modellen e yöntemlerinin bütün duru mlarında yankılanma göster nektedir. Uygulama alanlarındaki problemler özniteli k de ğ erlerinden ba ğ ı nsı z özel bir k i m i ğ e sahip olup ol ma dı ğ ıdır. Bu bir örnekle açıklanabilir. İsmi “ M i ke” olan birisi, isi m de ğ i l dir. İsi m de ğ i Ŗebilir veya vücutta bazı de ğ i Ŗi k li kler olabilir, fakat ki Ŗi ye ait k i m i k her zaman aynı kalacaktır.

Nesne k i m i ğ i sistemlere talepler do ğ rultusunda uygulanabilir. Her sistem özel bir obj e k i m i kleyiciye sahiptir ve asla bir obj e ol u Ŗ turul urken de ğ i Ŗ tirilemez ve sadece obj e tahrip ol mu Ŗ sa atılır. Bu k i m i kleyici di ğ er obj elerin özniteli klerine eklenir. Bir obj e kendi herhangi özniteli ğ inden ba ğ ı nsı z i ç tanı m h a ması na sahiptir.

Obj e k i m i ğ i ile ilgili olarak bazı bakı Ŗ açıları Ŗ öyledir.

- Bir obj e sistemi obj eler takı m olarak tanı mlanabilir.
- Obj e k i m i ğ i ne ek olarak kar ma Ŗ k obj elere ait kavramlarda, obj e kar Ŗıla Ŗ tır m a sı verilebilir.

1. *K i m i k*: İki obj e sadece ve sadece, özde Ŗ ol maları durumunda aynı obj elerdir.

2. *Sı ğ e Ŗ itlik*: İki obj enin e ğ er sadece özniteli kleri özde Ŗ ise, sı ğ e Ŗ itlik kavramından söz edilir.

3. *Deri n e Ŗ itlik*: De ğ erleri tekrarlanan iki obj e arasında deri n e Ŗ itlik bulunmaktadı r.

- Varlı ğ a ait destekleme, güç lü ve zayıf ol m a sı na göre ikiye ayrılmaktadır. K i m i ğ i n desteklenmesi i ç i ni ki farklı boyut vardır.

1. Sunum boyutu

2. Geçi ci boyut.

Sunum boyutunda en zayıf k i m i k desteklemesi bir obj enin veri tarafından k i m i klenmesi sonucu oluşur. Buna örnek olarak ilişkisel sistemlerdeki anahtar de ğ eri verilebilir. En güç lü destekleme ise k i m i ğ i i n Ŗ a et m ektir. Buna örnek olarak nesneye dayalı sistemler verilebilir.

- Bütün bunlara ek olarak obje paylaşım ve obje güncelleme adı verilen önemli bir konu daha vardır.

Obje paylaşımın anlamı iki veya daha fazla kaynak aynı objeyi gösterebilir şeklindedir. İki veya daha fazla obje aynı objeyi paylaşabilir. Örnek olarak iki obje (anne ve baba) diğer bir objeyi (çocuk) işaret edebilir. Bunlar bir objeyi paylaşır. Çocuk objesi üzerinde yapılacak olan bir güncelleme işlemi diğer iki objeye de yansıtacaktır. Bu yüzden sadece bir güncelleme yeterli olacaktır. Çocuk objesinin aile objeleri içine saklanması durumunda ise iki güncelleme işlemi gerekmektedir [13, 16].

#### **2.7.4 Depolama (Encapsulation)**

Yapısal programlama uygun olarak geliştirilmiş olan C dilinin en önemli problemi, bu dille yazılmış olan programın her kisinin her kullanıcı tarafından istenildiği zaman istenildiği biçimde değiştirilebilmesidir. Bu durumda kullanım sırasında tutarlılık ilkesinin bozulma ihtimali vardır. Bunu önlemek için C++ dilinde depolama (encapsulation) mekanizması geliştirilmiştir. Burada ifade edilmek istenen, bir objeye ulaşımın programlama sırasında tanımlanacak bazı işlemler ile sınırlı tutulmasıdır.

Kullanıcının çoğu kez veri modellemesinin içerdiği operasyonların ayrıntısını bilmesine gerek yoktur. Bir anlamda depolama (encapsulation) bazı işlemlerin kullanıcıya kapatılması olarak tanımlanabilir. Kullanıcının bu işlemlerin program tarafından nasıl yapıldığını bilmesine gerek yoktur. Onun için önemli olan saklı işlemlerin yazılım tarafından doğru yapıyor olmasıdır. Böyle bir mekanizma kullanıcıyı ayrıntılardan kurtarır ve veri tabanında tutarlılık sağlar. Bir nesnenin iç verisini ve metotlarını saklaması olanağı olarak tanımlanabilen veri gizleme sayesinde nesnenin yalnızca programlanabilir kısma erişilebilir.

Bazı obje grupları (sınıf) ve prosedürler (yöntem) veri saklama ve soyutlama prensibinde gizlenir ve birlikte yönetilir. Prosedürü aktif hale getirmek için program gizlenmiş veri-prosedür takımına bir mesaj yollar, sonuçta aktif olan prosedür farklı bir takım mesaj yollayabilir.

Soyut bir veri tipi, modülün bir parçasında gizlenir. Arayüz ise bu gizlenen veri ile çalışmayı destekler. Objeler, durum ve davranışları gizler. Nesneye dayalı sistemler ile çalışmalar yapanların herhangi bir veri saklama ve soyutlama derecesi bulunmamaktadır. Operasyonların arayüzler gibi destekleneceğini, yoksa özneliklere objenin dışından erişilebileceğini soruları nesneye dayalı sistemlerde sorulabilmektedir. C++, bu kararı kullanıcıya bırakır.

Prosedür gizlenmesi durumunda bir objeye ulaşabilmenin tek yolu prosedürlerdir (yöntemler). Diğer taraftan veri ve yöntemler özel veya genel olabilir. Bu durumda dış gizlenmeden söz edilir.

C++ ile esnek bir veri saklama ve soyutlama prensibi (encapsulation) yaklaşım amaçlanmaktadır. Kullanıcı özel, korunmalı, genel veri üyeleri ve üye fonksiyonları tanımlayabilmektedir. Özeli anlam, sınıfın sadece üye fonksiyonları için görülebilir olmasıdır. Korunmanın anlam, sınıfın ve onun alt sınıflarının üye fonksiyonları için görülebilir olmasıdır. Genel anlam, herkes tarafından görülebilir olmasıdır.

Veri saklama ve soyutlama prensibi olan encapsulation bir sistem yapısı ve aynı zamanda programlama felsefesidir. Gizlenen veri yasal girişler ile kısıtlanır. Yasal olmayan girişler veri tarafından engellenir ve üye fonksiyonlar tarafından desteklenir.

Objelerin durumları ve yöntemleri gizlenir. Daha yüksek bir seviyede tasarımı alt sistemleri veya uygulama modülleri diğer modüllerin arayüzlerine karşı bir genel arayüz tarafından gizlenir. Bu yüzden katmanlı yazılımlar bağımsızlığın ve uygulamanın bir seviyeden diğer sistemde tabakanın semantikliğinin etkilenmesini sağlar.

Nesneye dayalı programlama da, karmaşık veri tipi ile çalışılabilirliği, üst sınıfların oluşturulabilirliğini, kendine ait değişkenler içerisinde veriler tutabilen obje modelleriyle, ve tanımlanmış yöntemler kullanarak verinin değiştirilmesine olanak sağlanması veri depolama (encapsulation) sayesinde mümkün olmaktadır [ 3, 13, 16, 17, 20 ].

## 2.7.5 Otonomik Öznitelik Ve Fonksiyon Üstlenme ( Miras )

Nesneye dayalı veri modellenmesinde objelerin davranışları da veri olarak ele alındığına göre aynı davranışı gösteren birden çok objenin bulunması doğaldır. Örneğin taşınmazlar genellikle alınıp satılabilen objelerdir. Bazı taşınmazlar için bu satışı sınırlayan işlemler devreye girebilir. Yapısal programlama dillerinde bu yapıyı ifade etmek için araçlar yeterli uygunlukta değildir. Objelerin sınıfları genellikle birbirleri ile hiyerarşik bir ilişki içinde bulunmaktadır.

Burada hiyerarşideki alt sınıf, süper sınıfa ait bütün özellikleri miras alır. Aynı zamanda miraslanan bu özellikler dışında bazı ek özellikler de eklenebilir. Bütün bunlara ek olarak bir alt sınıfa geçen süper sınıfın bazı özellikleri fonksiyonluk veya gelişimi için başka bir davranış üretecek biçimde tekrar tanımlanabilir.

Objelerin hiyerarşisini anlatılması için “bina” örneği verilebilir. Binaları “resmi binalar” ve “resmi olmayanlar” olarak iki farklı parçaya ayırabilir. Bunlarda kendi içlerinde kullanım durumlarına göre resmi binalar “cami, okul” vb. resmi olmayanlar ise “konut”, “işyeri” vb. olarak farklı bölümlere ayrılabilir. Yine bir okul objesi ilköğretim lise ve meslek lisesi olarak ayrılabilir. Buna göre bina temel objedir. Binaya ait kapı numarası, yapı yılı, inşaat türü gibi özellikler öznitelik olarak modellenabilir. Bunun sonucu olarak resmi binalar ve resmi olmayan binalar hiç bir belirtme olmadan bina objesinin öznelikleri ne sahip olurlar. Öznelikleri değerleri farklı olabilir. Burada değer ve öznitelik kavramı karıştırılmaması gereken noktadır. Aynı zamanda lise objesi kendine ait ek özelliklere de sahip olabilir. Buna örnek olarak müdür ve spor salonuna sahip olması verilebilir. Bu özelliği sayesinde nesneye dayalı veri modellenmesi veri tutarlılığını güvenceye almakta ve kodlamada tasarruf sağlanmaktadır.

Miras önemli bir sistem ve gerçek dünya yapısıdır. Tekrar kullanım burada motive edici faktördür. Sistemlerin içeriğinde düzenlenmiş daha önceden var olan nesne veya obje tipinden yeni bir obje çeşidi veya tip oluşur. Gerçek dünyanın modellenmesinde miras iki farklı alanda görev alır. Bir diğerini tersidir.

- Genelleştirme
- Özelleştirme ( uzmanlık )

Genelleştirme:

Çeşitli nesne tiplerinin özelliklerinin soyutlanması. Örnek olarak “köy”, “şehir” verilebilir. Bunlar “yerleşim biriminin” açılımıdır. Yerleşim birimi burada bir genelleştirme dir.

Özelleştirme:

Nesne tiplerinin rollerine göre parçalanması. Örnek olarak yolcunun özeli olarak otoban kullanıcısı ve demiryolu kullanıcısı verilebilir.

Nesneye dayalı modeller bu yapıları farklı destekler. Öznitelik mirası, yapısal modellerin genişletilmiş olan EER dilinin özelliğidir, bununla birlikte nesneye dayalı kurulumlar da miras yöntemlerini desteklemektedir. Bu yüzden üçgen ve dikdörtgen, poligonun alt çeşitleri olarak ele alınabilir. At tipi süper tipin bütün öznitelik ve operasyonlarını miras alır.

Örneğizdeki üçgen ve dikdörtgen bazı operasyonlar ile özelleştirilebilir. Bu şekiller için eşkenar üçgen mi?, kare mi? gibi bazı özellikler sunulabilir. Boolean objeler eşkenar üçgen mi? Yoksa kare mi? olduğuna karar verir. Uygulama operasyonlar için kendilerinin de bazı özel yöntemleri olabilir ( çok biçimlilik ). Örneğin algoritma uygulaması “alan” operasyonu dikdörtgen veya üçgende farklı olabilir aynı zamanda, poligon yönteminde de farklı olabilir.

Miras iki farklı alana ayrılabilir. Bunlar tekil miras ve çoklu miras olarak adlandırılmaktadır.

### **2.7.5.1. Tekil miras:**

Bir sınıf sadece bir ataya sahiptir. Bir aile bulunur. Hiyerarşi sadece bir ağaca sahiptir. Özelleştirilmiş bir obje tipi tekil üretken tipin davranışlarının ve özniteliklerini miras alır. Bazı durumve davranışları kendisine eklemek mümkündür. Örneğin “görünen çoklu çizgi tipi”, “çoklu çizginin” özelleşmiş halidir. Bazı yöntemler eklenerek çoklu çizginin harita üzerinde veya ekran üzerinde görülmesi sağlanır.

### **2.7.5.2 Çoklu miras**

Burada birden fazla aile vardır. Tekli miras düşüncesinin genişletilmiş biçimidir. Burada bir tipin birden fazla süper tipten davranış ve öznelik alınmasına izin verilir. Çoklu miras daha karmaşık sistem desteğini gerektirmektedir. Bir alt tipteki bir operasyon için yöntemler birden fazla süper sınıftan miras alınabilir. Diğer bir çatışma ise, miras alınan öznelikler veya operasyonlardaki isim karmaşasından dolayıdır. Bu problemleri çözmek için bazı protokoller kullanılması gerekir.

Ayrıca semantik miras ve sintaktik miras da tanımlanabilir. Semantik miras, bir tasarım yapısıdır. Objeler arasında bir kavramsal bağımlılık uygulanır. Sintaktik miras ise programlama tekniğidir ve kodların tekrar kullanılabilirliğinde kullanılır.

Nesneye dayalı programlama dillerinde sınıflar arasındaki operasyonların paylaşımının desteklenmesi için miras çok sıklıkla kullanılır.

$$ADT \approx \text{tip} + \text{operasyonlar} + \text{veri depolama (encapsulation)} \quad (23)$$

$$\text{sınıf} \approx ADT + \text{miras} + \text{örneklerin koleksiyonu} \quad (24)$$

Miras kavramı ise aşağıdaki gibi tanımlanabilir;

Öncelikle “meyve” adlı özel bir sınıftan bahsediyor. Daha sonra üst sınıfa bağlı olarak “elma” adlı alt sınıfı üretmektedir. Yapılan hiyerarşik düzene göre “elma” alt sınıfı “meyve” üst sınıfının özelliklerine sahip olmasıyla aynı sıra kendine de ait birtakım özellikler de almaktadır.

// sınıf mirasına ilişkin bir örnek

```
#include <iostream>
```

```
#include <string>
```

```
namespace std kullanılıyor;
```

```
enum yn {hayır, evet};
```

```
enum color {kırmızı, sarı, yeşil, turuncu};
```

```
void out (enum yn x);
```

```
char *c[]={"kırmızı", "sarı", "yeşil", "turuncu"};
```

```
// Genel meyve sınıfı.
```



```

Class meyve{
// Temel sınıf içindeki herşey public
public:
    enumyn sürekli;
    enumyn mevsimlik;
    enumyn ağaç;
    enumyn tropik;
    enumyn tropik;
    enumcolor clr;
    char isim[40];
};
// Hma sınıfı üretiliyor.
Class Hma : public meyve{
    enumyn pişirmek için;
    enumyn suyu için;
    enumyn yemek için;
public:
    void sete (char *i, enumcolor c, enumyn ck, enumyn crchy, enumyn e);
    void göster();
};
void elma:: seta(char *i, enumcolor c, enumyn ck, enumyn crchy, enumyn e)
{
    strcpy (isim i);
    sürekli =hayır;
    mevsimlik=evet;

    ağaç=evet;
    tropik=hayır;
    clr=c;
    pişirmek için=ck;
    yemek için=crchy;
    yemek için=e;
}
void elma:: göster()

```

```

{
    cout << "İsim: " << n << "\n";
    cout << "Sürekli: "; out (sürekli);
    cout << "Mevsi: "; out (mevsi);
    cout << "Ağaç: "; out (ağaç);
    cout << "Tropik: "; out (tropik);
    cout << "Color: " << c << "\n";
    cout << "Pişirilir mi? "; out (pişirilmek için);
    cout << "Ezilip suyu çıkarılır mı? "; out (ezilmek için);
    cout << "Yenir mi? "; out (yenilmek için);
    cout << "\n";
}
void out (enum x)
{
    if(x == no) cout << "Hayır\n";
    else cout << "Evet\n";
}
int main()
{
    E e1;
    e1.set("Anasya", kırmızı, hayır, evet, evet);
    e1.göster();
    return 0;
} [ 3, 13, 17-19 ].

```

### 2.7.6 Çok Biçimlilik (Poli morfozluk)

Bir fonksiyon değişik sınıflardaki objelere uygulanmasıyla aynı mantıksal sonucu vermesi anlamındadır. Başka bir deyişle, aynı isimdeki operasyonların, değişik objeler için farklı biçimlerde algılanması olarak tanımlanabilir.

Çok biçimlilik (poli morfozluk) genetik bilimlere ait bir kavramdır ve genetik farklılıklar anlamındadır. Yazılım mühendisliğinde ise çok biçimlilik (poli morfozluk), nesneye dayalı veri modellerinin bir dizi özelliğini ifade etmek için kullanılır. Örneğin alt obje sınıfları için fonksiyonların (fonksiyon adları aynı kalması)

halde) yöntemleri farklılaştırma (overriding) ya da programlama dilinin standart operatörlerini (+,/,\*) yeniden tanımlanabilmesi nesne bindirme (overloading) gibi. Burada obje sınıfına uyan yöntemler (operasyonlar) otomatik olarak aranmakta ve belli bir zamanda ilgili objeye hangi yöntemin uygulanabileceğine karar verilmektedir. Örnek olarak bir kamu kurumunun bakım sorumluluğunda olan binalar için, binalar sahip oldukları öznelikleri (yapı türü, yıpranma derecesi, bulunduğu konum) bakımından farklılık göstereceklerdir. Diğer bir anlamda binalar kendi aralarında homojen olmayan özellikler taşırlar. Prensipte yapılacak işin karakteri aynıdır, fakat yürütülecek işlemler ayrıntıda şüphesiz farklı olacaktır.

Örneğin a, b, c adreslerine çizimi için bir mesaj gönderilirse obje içerisinde daire prosedürü içinde gizlenmiş olan a, elips prosedürü içindeki b ve kare prosedüründeki c adreslerine dayanarak çizim sonucunda daire, elips veya kare olarak ortaya çıkacaktır. Ya da başka bir örnek olarak üçgenin hesaplanması için kullanılacak yöntemlerin farklı olması örneğin hesaplama komutu aynı olmasına rağmen bir üçgenin hesaplanmasında koordinatlardan, kenarlardan veya üçgene ait açı ve kenarlardan faydalanılabilir. Bütün bunlara gönderilen komut aynı olmasına rağmen hesaplama işlemleri farklı olacaktır.

Çok biçimli (polimorfik) dillerde bazı değer ve değişkenler birden fazla tipe sahip olabilir. Dört çeşit çok biçimlilikten biri verilmektedir. Önce evrensel polimorfizm ve geçici polimorfizm olarak ilk etapta iki farklı polimorfizmden söz edilebilir. Buna göre evrensel polimorfizm, bazı değerlerin farklı tiplere sahip olabilmesi durumudur. Uygulamada aynı kodlar farklı tipler için kullanılabilir, şeklinde bir anlatım yapılabilir. Parametrik ve kalıtı polimorfizmi adı altında evrensel polimorfizmin iki alt tipi bulunmaktadır. Buna göre parametrik polimorfizm üretken (generic) fonksiyonlar olarak anlaşılır. Kalıtı polimorfizmi ise miras içinden ayrılır. Bir obje farklı sınıflara (süper sınıflar, alt sınıflar) ait olarak görülebilir.

Geçici polimorfizmde ise, farklı kodlar farklı tipler için kullanılır. Geçici polimorfizmin iki alt üyesi ise nesne bindirme (overloading) ve zorlama (coercion) dir. Nesne bindirme de aynı operasyon aynı inanza ile farklı yöntemlere uygulanır. Zorlama (coercion) ise keskin dönüştürülmüş tiplerdir (tam sayıdan gerçek sayıya gibi). Zorlama (coersion) C ve C++ da bir kalıptır.

Aynı operasyonun, farklı yöntemler ile farklı sınıflara uygulanabilmesi işlemi olan polimorfizm için bir örnek daha verilebilir: Örneğin çevre hesabı kare, üçgen ve daire sınıfları için farklı uygulamalara sahip olabilir. Polimorfizm özellikle miras kombinasyonu ile çok güçlü hale gelen bir araçtır. Böylece bilgi sistemlerindeki yöntemlerin esnekliği desteklenir [ 3, 13, 16, 20 ].

### **2.7.7 Yöntemlerin Farklılaştırılması (Overriding)**

Alt sınıfta tanımlı olan bir operasyon, bir süper sınıfta da aynı işi yapabilir. Bu örnekte ‘göster’ operasyonu hem ‘öğrenci’ sınıfında hem de ‘öğrenci’ sınıfının üst sınıfı olan ‘kişi’de tanımlanmıştır. Yöntemleri farklılaştırmaya (overriding) örnek olarak aşağıdaki durum verilebilir.

Eğer bir sistem overridingi destekliyor ise, alt sınıftaki bir operasyon o sınıfın doğrudan veya dolaylı olarak üst sınıfının içinde bir veya birden fazla operasyon olarak aynı işi yapabilir. Alt sınıftaki operasyondan süper sınıftaki operasyonu override etmesi talep edilir.

C++ fonksiyonları aynı adı taşıyan birden fazla fonksiyon kullanımına izin vermez. Burada önemli olan nokta, aynı adı taşıyan fonksiyonların parametre sayılarının ve/veya tiplerinin farklı olması gerektiğidir [ 15, 18 ].

### **2.7.8 Nesne Birleştirme (Overloading)**

Nesne birleştirme (overloading) olduğu zaman, sistem aynı işi yapacak özelliklere sahip farklı sınıflar olabileceğini mümkün kılar. Buna örnek olarak şöyle bir örnek verilebilir. Bir ‘kişi’ sınıfı ve ‘bina’ sınıfının her ikisi de ‘yükseklik’ özneliğine sahip olabilir. C++ dilinde, dilin içinde tanımlı olan ve yinelenmesine izin verilmiş operatörler, farklı görevler alabilirler. Operatör yinelenmesiyle, bu operatörler objeler üzerinde çalıştırılabilirler [ 15, 18 ].

### 3. VERİ TABANI YAPILARI VE VERİ TABANI YÖNETİMİ

Bilgi sistemlerinde veriler veri tabanlarında depolanmaktadır. Veri tabanı farklı uygulamalarda ortak olarak kullanılabilen, yinelenenlerden arındırılmış ve ilişkilendirilmiş verilerin oluşturduğu veri kümesidir.

Gerçek dünyadaki varlıklar basit veya organize edilmiş listeler halinde veri olarak bilgisayar dosyalarında saklanmaktadır. Bilgisayar daki bir dosyanın temel ünitesi veri kayıdır. Veri kaydı, her varlığı için varlıkla ilgili olan bütün bilgileri içerir. Veri kayıtlarının elde edilme biçimine bağlı olarak değişik uzunluklarda veri kayıtları bulunur. Varlığa ait veriler mekansal özelliklerinden dolayı birlikte gruplandırılması gerekmektedir. Bu tür olayların organize edilmesinde anlatılan yol çok önemlidir. Dosyalar bilgisayarda gerçek dünyaya ait olayın modellenmesindeki gibi ilişkilendirilir ve saklanır. Böylece elverişli saklama ve verinin yeniden kullanımı sağlanmış olur. Bir bilgisayar programı büyük miktarlardaki verinin saklanması ve yönetilmesi için tasarlanır. Uygulanan sistemler veri tabanı yönetimsistemleri olarak adlandırılır.

Moder n veri tabanı yönetimsistemleri yeterli saklama ve verinin yeniden kullanımı için pek çok yöntem kullanır. Fakat bütün bunların hepsi organize edilmiş üç temel bilgi çeşidine bağlıdır.

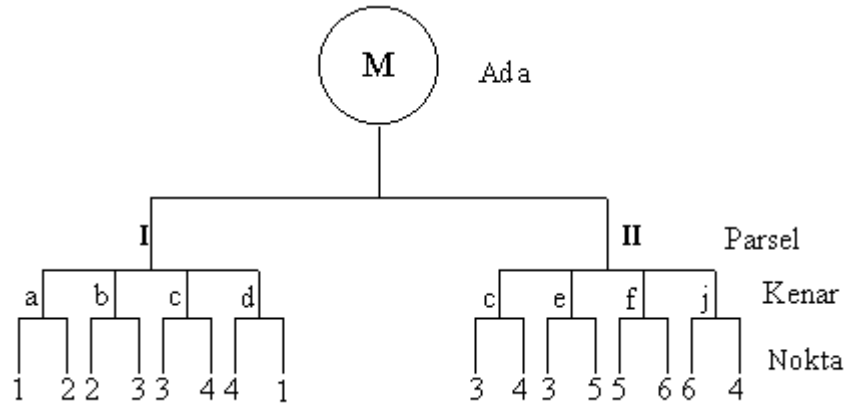
- Hiyerarşik
- Ağ
- İlişkisel veri tabanı

Bütün bu veri tabanları CBS lerde kullanılmaktadır. Fakat yakın zamanlarda 4. bir yapıda CBS de kullanılmaktadır. Bu yapı, “nesneye dayalı”dır. Nesneye dayalı veri tabanları sayesinde mekansal varlıklar arasındaki sahiplik ve bağlantılar daha etkili bir biçimde modellenmektedir. Nesneye dayalı çalışmalar bazı ticari CBS alanlarında kullanılmaktadır, fakat kavram üzerindeki çalışmalar hala devam etmektedir.

Veri tabanı modellenmesini n asıl amacı, veri tabanının yeterli performansta, doğru bilgiler içeren şekilde tasarlanmasıdır. Aşağıda adı geçen veri tabanlarına ait bilgiler verilmiştir [ 5, 21 ].

### 3.1 Hiyerarşik Veri Tabanı Yapıları

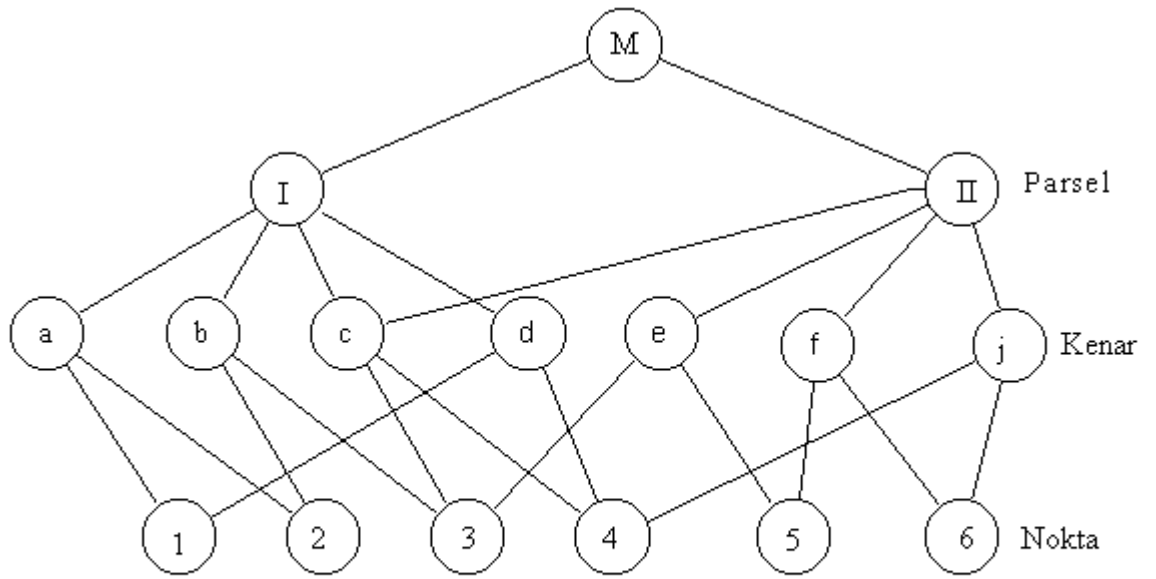
İlk olarak 1968 de IBM tarafından geliştirilmiştir. Hiyerarşik veri modeli ağaç yapısındadır (Şekil 3.1), veriler arasında aile/çocuk ya da bire-çok ilişki vardır. Çok-a-çok ilişki doğrudan sağlanmaktadır. Nesnelere arasındaki ilişkiler uygulama programlarında kodlanır. Şekilde hiyerarşik veri yapısında her nesne için kodlanmış veri kayıtları gösterilmiştir. Ada, parsel, kenar ve nokta içeren dört nesne arasında hiyerarşik bir ilişki vardır. Hiyerarşi düzenlenmesi ve erişim amacıyla anahtar alanlar/öznitelikler tanımlanmıştır. Hiyerarşinin en üst düzeyine kök denir (ada). Kökün dışında her nesne üst düzeyde yalnızca aile denilen tek bir nesneyle ilişkilidir, altındaki nesnelere de çocuklar denir. Her nesnenin yalnız bir ailesi, çok sayıda çocukları olabilir. Daha önce sözü edilen bire-çok ilişki örneklenirse, bir adada pek çok parsel olabilir, her parselde pek çok kenar olabilir. Bilgiye ağaç yapının izlenmesi ile ulaşılabilir. Belirli bir parseldeki tüm kayıtlara erişmek aralarında ilişki olduğu için kolaydır. Ancak bir parseldeki tüm noktalara erişmek için 2 aşamalı arama yapılır. Hiyerarşik sistemlerde yapıyı anlamak ve güncelleştirmek, anahtar özellikleri ile kolaydır, ancak birleştirilmiş özniteliklere erişim çok zordur. Erişim güçlüğü nedeniyle veri tekrarı da söz konusudur. Günümüzde CBS yazılımlarında kullanılmaktadır.



Şekil 3.1: Hiyerarşik veri modeli

### 3.2 Ağ Veri Tabanı

Ağ veri modeli ağaç yapısına benzer ancak her çocuğun birden çok ailesi olabilir, nesnelar arasında bire- çok ve çoka- çok ilişki söz konusudur (Şekil 3.2). İstenilen kayıta, köke uğramadan erişilebilir. Hiyerarşik modele göre veri tekrarı azdır, daha esnektir, ancak daha karmaşık yapıya sahiptir. Her nesne öz nitelikleri ile birlikte ağda bir düğüm noktası olarak düşünülür. Nesnelar arasındaki ilişkiler için nesnelar arasında göstergeler (pointers) kullanılır. İşaretleyici bilgileri veri dosyalarında tutulur [ 5].



Şekil 3.2: Ağ veri modeli

### 3.3 İlişkisel Veri Tabanı

Hiyerarşik ve ağ modellerinde en büyük problem veriler arasındaki ilişkilerin veri tanımlaması aşamasında kurulmuş olmasıdır. Bazı uygulamalarda veri ilişkisinin dinamik olması istenir. Codd [5] tarafından 1970' de geliştirilen ilişkisel model, veriler arasında ilişki kurmada kullanıcıya büyük esneklik sağlamaktadır. Mantıksal veri modellemek için ağaçlar, ağlar yerine ilişkileri gösteren satır ve sütunlardan oluşan tablolar kullanılır (Şekil 3.3). Yeni ilişkiler kurmak için veri yapısı bozulmaksızın ilişkisel cebir veya ilişkisel hesaplama teknikleri kullanılır. Sorgulamada

işlemleri sırasında da çok kullanışlıdır. Nesnelar arasında bire-bir ilişki vardır. Çok-çok ilişki için ara tablolar hazırlamak gereklidir. Her nesneye ait öznitelikler bir tabloda tanımlanır.

Şekil 3.3 ilişkisel veri modelinde tasarlanmış tabloları ve kayıtları göstermektedir. Tablolardaki kayıtlara erişmek için değeri tek olan anahtar alanlar/öznitelikler kullanılır. Örneğin parsel tablosunda kenar özniteliği, kenar tablosunda kenar özniteliği ile birleştirilebilir. Örnekte Ada-Parsel, Parsel-Kenar tablosu arasındaki ilişki Parsel No özniteliği ile, Parsel-Kenar ve Kenar-Nokta ilişkisi ise Kenar özniteliği ile kurulur. Sorgulamada anahtarları yardımıyla Madasında bulunan kenarları oluşturulan noktalara da erişilebilir.

Ada no	Parsel No
M	I
M	II

Parsel no	Kenar
I	a
I	b
I	c
I	d
II	c
II	e
II	f
II	j

Kenar	Baş. No	Bit. No
a	1	2
b	2	3
c	3	4
d	4	1
e	3	5
f	5	6
j	6	4

Şekil 3.3 İlişkisel veri modeli

İlişkisel veri tabanı modeline göre tasarlanmış programlarla çalışmak, sistemin kuruluşu ve kullanımı ve değişiklikler yapılması açısından diğerlerine göre daha kolaydır. Ancak bir sorgulamada için bir çok tablonun ilişkilendirilmesi, yanıt süresini uzatabilmektedir. Ağ modeline göre veri tekrarı daha çok olabilmektedir. ORACLE, INFORMIX, dBASE vb. ilişkisel modele göre tasarlanmış ve kodlanmış veri tabanı yönetim sistemleri günümüzde CBS ortamında yaygın olarak kullanılmaktadır.



### 3.4 Nesneye Dayalı Veri Tabanı

Nesneye dayalı veri tabanı yapıları nesneye yönelik programlama dilleri kullanılarak geliştirilmiştir. Buna ek olarak hiyerarşik yapının ve ağ yapının hızı birleştirilmiş ve ilişkisel yapının esnekliği ilave edilmiştir.

Nesneye dayalı veri tabanlarında veri, özel obje serileri içinde tanımlanır. Bunlar aynı doğal olayları göstermelerine göre organize edilirler, bu da obje sınıfları adıyla bilinir. Objeler arasındaki ve farklı sınıflar arasındaki ilişkiler kesin bağlantılar ile kurulur.

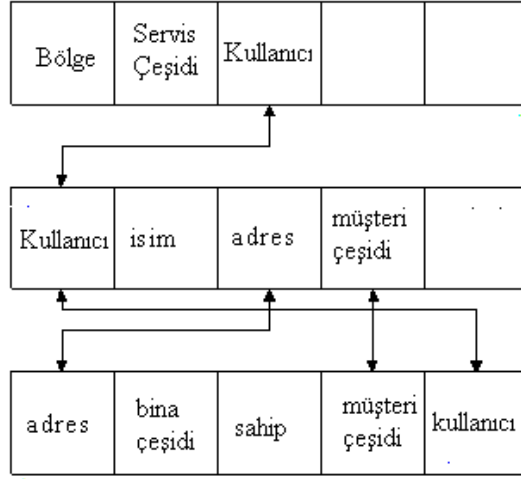
Objenin özellikleri veri tabanında öznitelikler veya durumlar adıyla tanımlanır. Aynı zamanda prosedürlerin de çok iyi tanımlanması gerekir. Bunlar davranışları (operasyonlar veya yöntemler olarak da adlandırılır) tanımlamak için kullanılır. Veriler obje içerisinde gizlenir.

Objeye veri tabanında özel bir kimleyici tarafından tanımlanır. Kalıntılar ise değişkenlikler ne olursa olsun aynıdır. Aynı değerler değişir, ama tanımlanan özellikler aynıdır. Örneğin bir binanın objesinin zaman geçtikçe yapısı veya kullanımı değişebilir. Fakat özel kimleyicisi aynı kalacaktır.

Veri tabanındaki objelerin yapısı göstergeler kullanılarak kurulur. Bu göstergeler özel kimlikleri doğrudan sunarlar.

Sınıflar ve örnekler çeşitli ilişkileri ve hiyerarşileri göstermede göstergeler üzerinden ilişkilendirilir. Alt sınıf, üst sınıflar, çeşitli tanımlı durumlar veya yöntemler miras sistemi boyunca oluşturulur. Bu objelerin özniteliklerinin karakterize edilmesi de veri tabanından tekrar kullanımlarında yapılabileceği anlamına gelmektedir.

(Şekil 3.4) ilişkisel veri tabanı ve nesneye dayalı veri tabanını karşılaştırmaktadır. Burada ilişkisel sistemde veriler bir tablodan diğerine geçerken tekrar tanımlanmışlardır.



Şekil 3.4 İlişkisel veri modeli gösterimi

Nesneye dayalı veri modelinde (Şekil 3.5) ise , bölge, kullanıcı ve adres objeler olarak tasarlanmıştır. Bunlar arasındaki ilişkiler örneğin “içerir”, “tarafından kullanılır” gibi tasarlanmıştır. Bütün veri sadece bir defa tutulur ve doğrudan göstergeler sadece veriye erişimi kolaylaştırır, aynı zamanda veri tabanı hiyerarşisi sırasında aşağı yukarı komut geçişi sağlar.

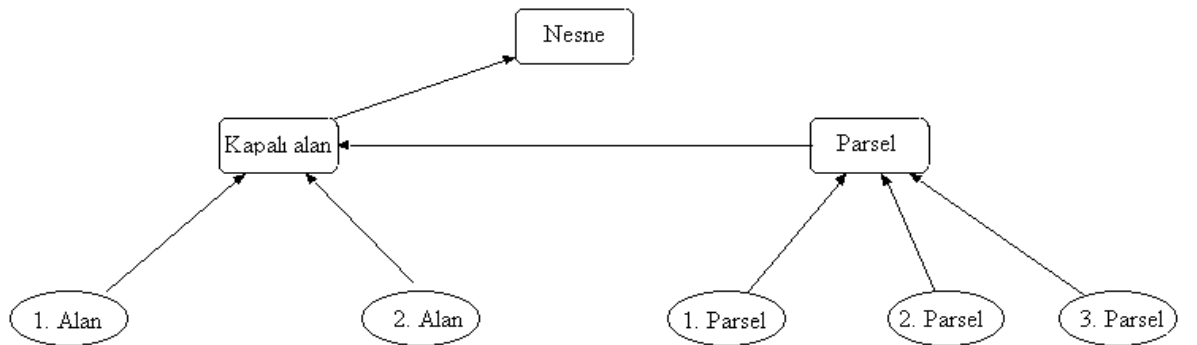


Şekil 3.5: Nesneye dayalı veri modeline göre aynı işlemin gösterilmesi

Eğer bir veri bir veri tabanında bir nesne içine depolanmış (encapsulasyon) ise veriyi değiştirebilmenin veya sorgulayabilmenin tek yolu bir istek göndermektir. Bu istek mesaj olarak adlandırılmaktadır. Bu operasyonlar tarafından gerçekleştirilir. Sorgulamaya çeşitli operasyonlara bağlı olabilir. Operasyonlar objenin tanımlanmasında kullanılmışlardır. Nesnenin mesajı yanıtı, objenin durumuna bağlıdır. Farklı objeler veya farklı içerikler mesaj alındığında aynı mesaja farklı reaksiyon gösterilebilir. Bunun ise çok biçimlilik (polimorfizm) olarak adlandırıldığı daha önce belirtilmişti.

CBS uygulamalarının gerektirdiği veri yapıları karmaşıktır (geometrik ve öznelik veriler, analizler vb. nedeniyle). Bu yapıda ilişkisel model verinin bir çok ilişkiye bölünmesini, verilerin ayrı ayrı düşünülmesini gerektirmektedir. Bu durum yüzlerce ilişkiyi tablonun birleştirilmesini gerektirdiğinden performans düşüktür. 1990'lı yıllarda nesneye yönelik veri modeline dayalı teorik ve pratik anlamda yeni kuşak CBS'ler için çalışmalar başlamıştır. Ayrıca ilişkisel veri tabanı ortamında nesne yönetimini sağlamak içinde çeşitli teknikler geliştirilmiştir. Örneğin, ilişkisel veri tabanı tablolarının nesne düzeninde hazırlanması gibi. İlişkisel veri tabanı yönetim sistemine, nesneye yönelik programlama özellikleri eklenerek uygulamalar yapılmıştır. Intergraph TIGRIS yazılımında olduğu gibi baştan sona nesneye yönelik mantıklı CBS yazılımları da geliştirilmiştir.

Nesneye yönelik veri modelinde her nesne bir sınıftır. Nesne; veri ve verileri kullanacak fonksiyonları/işlemleri içerir. Ortak özellikleri olan nesnelere bir sınıfa aittir. Her sınıfın bir süper sınıf olabilir (Şekil 3.6). Süper sınıfın verileri ve işlemleri alt sınıflarına kalıtsal olarak geçer. Şekil 3.6 incelenecek bir nesne sınıfı olan "kapalı alan" sınıfı, diğer bir sınıf olan "parsel" sınıfının süper sınıfıdır.



Şekil 3.6 Nesneye Dayalı Veri Modellemesine At Bir Gösterim

Şekil 3.6 de yer alan “kapalı alan” ve “parşel” nesneleri aşıağıdaki şekilde tanımlanabilir.

<u>“Kapalı alan” nesne sınıfı</u>	<u>“Parşel” Nesne Tanımı</u>
<u>Süper Sınıflar (nesne)</u>	<u>Süper sınıflar (kapalı Alan)</u>
<u>Sınıf Değişkenleri</u>	<u>Sınıf Değişkenleri</u>
Kapalı alan no	<u>Öznitelikler</u>
<u>Öznitelikler</u>	Değer
Noktaları n listesi	Malik
Kenarları n listesi	<u>Fonksiyonlar</u>
Alan	Malik Değişimi
<u>Fonksiyonlar</u>	Ayırma
Hesaplama	
Çizim	
Üst üste karşılaştırma	

“Kapalı alan” süper sınıfının fonksiyonları ve öznitelik değerleri bu yapı bozulmaz ise “parşel” sınıfına kalıtsal olarak geçmektedir. Nesneye yönelik model ile hazırlanmış veri tabanı yönetim sistemlerine örnek olarak SMALLTALK, Object Store, O2, ORION verilebilir [ 5, 21 ].

**Object Store** veri tabanı yazılımı hakkında aşağıda kısa bir bilgi verilmiştir.

Object Design Inc tarafından geliştirilen Object Store tıpkı O2 gibi nesneye dayalı veri tabanı grubunda yer alır. Bir sayfa sunucusudur. Diğer programların tersine isteneni için veri erişimi sanal hafıza ortamında yapılır. Bu nedenden dolayı objeye erişim ana hafıza içinde yapılandırılmıştır. Böylece veriye erişimde olduğu kadar verideki geçişte aynı hızla yapılabilir. Object Store geniş kullanıcı ön belleği oluşturur. Geniş ölçekli uygulamaların yanı sıra Object Store obje sunumu için Java uygulamalarında yerel (tek kullanıcı) olarak geniş yaygınlıkta bir kullanım alanı sağlar. Object Store’un sınırlı bir versiyonu web sunucularla ( Netscape Communicator, Microsoft Internet Explorer) entegre edilmiştir.

### **3.5 Genişletme Kavramı**

Mekan referanslı terim dünya da mekansal olarak yeri bulunan coğrafi doğal olay ile ilgilidir. Mekan referanslı varlıklar yoğunluk gibi özel özellikleri ve diğer varlıklar

ile olan uzaklık gibi mekansal ilişkileri ile karakterize edilirler. Mekan referanslı verinin doğası gereği mevcut veri tabanı yönetim sistemleri araçları CBS operasyonlarını yeterli biçimde destekleyebilmeleri için genişletilmeye ihtiyaç duyulmaktadır [2].

### **3.5.1 İlişkisel veri tabanları**

Nesneye dayalı programlama teknikleri zorunlu olarak nesneye dayalı veri tabanlarını gerektirmez. Örneğin nesneye dayalı bir CBS olan SMALL WORLD 2.1 ilişkisel veri tabanı ORACLE kullanmaktadır [3]. İlişkisel veri tabanları öncelikle standart uygulamalar (mekansal veriler dışında) için geliştirilmiştir. Buna karşın çoğul ortamlar (Multi-Media) gibi uygulamalar standart dışı olarak adlandırılmaktadır. İlişkisel modelin sadece veri tabanı ile sınırlandırılmasının aksine, daha önceden bahsedilen nesneye dayalı kavram programlama dili, veri tabanı, yazılım mühendisliği veya herhangi bir şey ile sınırlandırılmamıştır. Sonuçta butür ihtiyaçları karşılamak amacıyla nesneye dayalı programlama dillerine paralel olarak nesneye dayalı veri tabanları geliştirilmiştir.

CBSlerde bugün ilişkisel veri tabanı yaygın olarak kullanılmaktadır. Bunun iki ana nedeni söz edilebilir.

1. İlişkisel veri modelinin matematik modeli açık olmaktadır.
2. İlişkisel veri tabanı modeli sorgulamaya dili olan SQL (Structured Query Language) tarafından kolay ve verimli olarak kullanılabilir.

Diğer taraftan ilişkisel veri tabanlarının özellikle planlama amaçlı CBSler için yeterli olmadığı konusunda görüşler ifade edilmektedir. Bunun önemli nedenleri, CBS sektöründe yönetilecek verinin hacminin büyümesi ve uygun planlama çözümleri için planlamacı, yönetici ve vatandaşı destekleyecek somut ve soyut varlıklara da ihtiyaç duyulmasıdır [3, 16].

### **3.5.2 En uygun veri tabanı yapısının seçimi**

Açıklanan dört veri yapısı da mekansal bilgi sistemleri için kullanılabilir olanlar sunmaktadır.

- Hiyerarşik yapıdaki sistem büyük veri tabanlarının yönetilebilir bölümler içerisine kolaylıkla bölünebilir olmasına izin vermektedirler. Fakat yeni araştırma yapıları eklemek için yeterli esneklikte değildirler. Çok fazla gereksiz bilgi de içerebilirler.
- Ağ sistemler daha az gereksiz veri ya da hiç gereksiz veri içermezler. Eğer varlıklar arasında esneklik olmayan bağlantılar varsa doğrudan erişim ve hız desteklerler.
- Nesneye dayalı sistemler fonksiyonluk, ilişki, devamlılık ve iç bağımsızlığa izin vermektedirler.
- İlişkisel sistemler esnek, uygulanabilir yapıdadırlar, fakat geniş veri değerleri, gereksiz veriler sonucunda etkisini yitirmekte ve zaman kayıplarına yol açmaktadırlar.

Sonuçta bütün yaklaşımların tek başlarına kullanılmaktansa birleştirilerek kullanılmaları daha uygun görülmektedir. Hiyerarşik sistemler; mekansal verilerin az olduğu, sürekli harita yapımı için, ağ yaklaşımı; vektör çizgilerin ve poligonların topolojik bağlarının kurulmasında, ilişkisel yaklaşımlar; var olan veriden yeni öznitelik ve öznitelik değerleri oluşturmada ve nesneye dayalı yaklaşımlar ise, varlıklara ait özniteliklerin paylaşımında kullanılmaları uygun olacaktır [21].

### 3.6 Veri Tabanı Yönetim Sistemleri

Verilerin veri tabanında depolanması, güncelleştirilmesi, erişilmesi veri tabanı yönetimsistemleri tarafından sağlanmaktadır.

Veri tabanı yönetimsistemleri veri tabanının yönetilmesi ve organize edilmesi için hiyerarşik, ağ, ilişkisel ve nesneye dayalı yapıların birlikte veya tek başına kullanılarak elde edildikleri bilgisayar programlarıdır. Veri tabanı yönetim sistemlerinin amacı verinin çoklu kullanıcılara hızlı ulaşması, verinin silinme ve bozulma ihtimaline karşı koruma ve verinin ihtiyaç duyduğu durumlarda güncellenmesi, taşınması ve eklenmesi gibi araçları uygulayabilmektir. Veri tabanı yönetimsistemlerini aşağıdaki fonksiyonları desteklemesi tavsiye edilmektedir.

- Verinin saklanması ve yeniden kullanımına izin verilmeli ve veri seçimi bir veya daha fazla özneliğe veya ilişkiye dayanmalıdır.
- Uygulama programlarında veriye standart erişim ve ayrık veri saklanması ve verinin kullanımının geri alınması bu programlarda bağımsızdır.
- Veri tabanı ve uygulama programı arasındaki arayüz desteklenmelidir.
- Uygulamadaki fiziksel saklama yapısının bağımsızlığı sayesinde medyadaki saklı programlar değişiklikler tarafından etkilenmez.
- Aynı anda veriye pek çok kullanıcı erişimi olabilmelidir.
- Yasal olmayan ve rasgele erişim tehlikesinden veri tabanı korunmalıdır.
- Sistem veri tutarlılığı için sesli uyarılar kullanılarak desteklenmelidir. Bu uyarılar hataların ve tutarsızlıkların veri tabanından uzaklaştırılması için etkili yoldur [ 5, 21 ].

### 3.6.1 Vektör ve raster veri yapılarının veri tabanında saklanması

Daha önceki bölümlerde anlatılmış olan veri tabanı yapılarının he men hepsi raster ve vektör verinin saklanması için kullanılabilir. Çoğu CBS lerde ilişkisel veri tabanı yapıları kullanılmasına rağmen günümüzde nesneye yönelik veri tabanları da karşımıza çıkmaktadır.

CBS tasarımcıları dosyalama sistemlerinin avantajlarına karşılık dijital harita koordinatlarını saklamak için hibrit veri modelleri geliştirmişlerdir. Sayısal mekansal veri, doğrudan erişilebilir operasyon sistem dosyalarında giriş ve çıkışta hızlı erişimi için saklanırken, öznelikler genelde ticari ilişkisel veri tabanı yönetim sistemlerinde saklanır.

#### Hibrit ilişkisel veri tabanları:

Uygun ticari ilişkisel veri tabanı yönetim sistemleri örneğin INFO, ORACLE, INGRES, INFORM X ve benzer ürünler CBS çalışma alanlarında oldukça yaygın olarak kullanılmaktadır. Bu veri tabanları tasarımcılara mekansal veri tabanı yönetimi ile ilgili olarak karşılaşılan problemleri iki parçaya bölünmesini sağlamaktadırlar. İlk kısım mekansal objelerin topoloji ve geometrisinin nasıl sunulacağıdır. Bu ancak vektör veya raster veri yapıları ile sağlanır. İkinci kısım ise mekansal objelerin özneliklerinin nasıl işleneceğidir. Bu ise ticari ilişkisel veri

tabanı yönetimsistemleri kullanılarak yapılabilir. Sonuçta oluşan hibrit veri yapıları bazı avantajlara sahiptir.

- Öznitelik verisinin mekansal veri tabanı ile saklanması na gerek yoktur, fakat sistemüzerinde veya on-line bağlantılı olarak herhangi bir yerde korunabilir.
- Öznitelik verisi genişletilebilir, erişim sağlanabilir, silinebilir veya güncelleştirilebilir, bütün bunlar mekansal veri tabanında düzeltme gerektirmez.
- Ticari ilişkisel veri tabanı yönetim sistemleri, yeni gelişmelerin sisteme entegrasyonunu garanti etmektedir.
- Veri yapıları, veri sözlükleri kullanılarak standart yollarla tanımlanabilir. Veriye erişimi ilişkisel veri tabanı yönetim sistemlerinden bağımsız olarak SQL gibi genel yöntemler ile yapılabilir.
- Bir ilişkisel veri tabanı yönetim sistemlerinde öznitelik verilerinin saklanması na CBS deki temel tabaka prensipleri ile müdahale edilemez.
- İlişkisel veri tabanı yönetim sistemlerindeki öznitelik verileri mekansal veriler bileşeni ile ilişkilendirilebilir ve sunumları çok farklı biçimlerde olabilir.

Ticari CBS lerde kullanılan ve CBS tasarımcıları tarafından geliştirilmiş bazı hibrit yapılarının isimleri aşağıda verilmiştir.

- Yay-düğü ilişkisel veri tabanı yönetim sistemleri
- Sıkıştırılmış raster ilişkisel veri tabanı yönetim sistemleri
- Dört ağaç (Quadtree) ilişkisel veri tabanı yönetim sistemleri
- Objel ilişkisel (object relational) veri tabanı yönetim sistemleri [ 21, 22 ].

### **3.6.2 Nesneye dayalı veri tabanı yapıları: Birleştirilen öznitelikler ve geometrik saklama**

Nesneye dayalı veri tabanları coğrafi verinin atomik bölüm serileri halinde tanımlanmasını talep ederler. Burada varlık kavramsal modeli kullanılarak veri tanımlaması bir kolaylıktır. Coğrafi veriler öznitelik ve davranış değerleri tarafından karakterize edilirler. Bu değerler verilerin mekansal, grafik, değişken ve metinsel/nümerik boyutlarını tanımlar. Öznitelik tanımının bir bölümü objenin geometrik doğasını (nokta, çizgi, poligon veya hücre) tanımlayacaktır.



Hiyerarşik ilişkiler çeşitli sınıflar ile kurulabilir. Örneğin yay objesi poligon objesinin alt sınıfı olabilir. Çeşitli obje örnekleri ve sınıflar arasındaki topolojik bağlantılar açıkça obje noktaları ve operatörleri boyunca kurulur. Örnek olarak “yön”, “kesişi mi”, “bitişi lik”, “örtüş me”, “solunda”, “sağında” gibi topolojik ilişkiler verilebilir.

Hiyerarşik yapılaş ma ve obje sınıfları arasındaki karmaşık ilişkilerin sunumu doğrudan kontrol edilebilir. Bu yüzden veri tabanının güncellenmesi ve değiştirilmesi esneklik kazanır [ 21 ].

### **3.6.3 Nesneye dayalı veri modellemesi yönetimi sistemlerinin özelliklerini n coğrafi obje yönetimi ndeki faydaları**

- Objeler tek bir birimde veri ve davranışları gizler (encapsul ation).
- Özel yöntemler, çok boyutlu erişim yöntemleri kullanılmak için bir objenin davranışlarına eklenebilir. Objelerin koordinatlarına dayanarak erişim yöntemleri buna örnek olarak verilebilir.
- Objeler özel kimliklere sahiptirler.
- Karmaşık objeler ve miras desteklenir. Örneğin “taşınmaz mal” parsellerden oluşur. Bir “taşınmaz mal” obje sınıfı bir parsel obje sınıfı ile toplam (agregasyon) ilişkilere sahiptir. Bir köy obje sınıfı ve kasaba obje sınıfı yerleşim obje sınıfının özelliklerini miras alır.
- Objelerin özellikleri karmaşık değerlere sahip olabilir. Diğer objelerin içi n kaynak teşkil ederler. Listeler, veri dizileri ve prosedürler bunlara örnek verilebilir. Poligonlar kenarlardan oluşur kenarlar ise noktalardan. Bir poligon obje sınıfı, kenar obje sınıfı ile düzenli ilişkilere sahiptir. Mesela kenar objelerini listesi öznelik olarak kabul edilir. Bir kenar obje sınıfı, poligon sınıfı ile ilişkilidir. Poligon obje takımları da burada öznelik olarak kullanılır. Aralarındaki ilişkinin kardinalliği m m dir. yukarıdaki anlatılan özelliklerin hepsi kenar ve nokta obje sınıfları için de geçerlidir. Aralarındaki kardinallik ilişkisi m m dir. Bir nokta obje sınıfı, içinde x, y, z koordinatlarının da bulunduğu çeşitli özneliklere sahiptir [ 23 ].
- Dolaylı bütünlük desteklenir. Bir yol silindiği zaman silinen yola ait bütün parçalar da beraberinde silinir.

- Görüntü gibi geniş veri blokları bir objenin öznitelikleri olarak saklanabilir. Örneğin bir evin resmi eve ait öznitelik gibi saklanabilir.
- Objelerin versiyonları ve veri tabanları konfigürasyonları eski bilgilerin korunması için kullanılabilir. Örneğin karışık obje gibi geniş bir veri grubu kilitlenebilir. Objenin yeni versiyonu güncelleme gerçekleştirildiği her zaman oluşturulabilir.
- Şema evrimi desteklenir. Var olan uygulamalarda az bir emek verilerek gerekli düzenlenmeyi yapmak mümkündür.
- Veri tabanı şemasını tanımlayan meta veri sorgulanabilir. Kullanıcının veri tabanının şemasını gelişmiş düzeyde bilme zorunda değildir.
- Tek kullanıcı performansı yeterlidir. Bazı CBS uygulamalarında, özellikle de planlamaya ilişkin olanlarında eş zamanlı erişim daha azdır. Veri tabanı yönetim sistemlerinde tek kullanıcı performansı oldukça önemlidir.

Bütün bunlara ek olarak nesneye dayalı veri tabanı yönetim sistemi geniş hacimli veri yapısının saklanması, sorgulama dili, uygulama biçimi paketleri, dağıtık veri tabanı yapılarını, tetikleyici işlemleri desteklemeyi, veri tabanını küçük birimlerini kilitleyebilmeyi, arıza toparlayabilmeyi ve veri bağımsızlığın desteklemek gibi geleneksel veri yönetim yeteneklerine sahiptir [ 23 ].

### 3.6.4 İlişkisel hibrit CBS ler ile nesneye dayalı yöntemlerin karşılaştırılması:

#### İlişkisel hibrit CBS

##### Avantajları:

- Mekanal verinin sisteme girildiği en sonra düzenlenebilmesi mümkündür.
- Veriye erişim ve fonksiyonluğun modellenmesi veri tabanı yönetim sistemleri tarafından desteklenir.
- Diğer sistemlerden veri tanımlanması oldukça kolaydır. Bu özellikle öznitelik verileri için daha kolaydır.
- Verinin bütün yönleri özelleştirilmiş dosya yapılarında saklanır.
- Kullanımı kolaydır.
- Ses alt yapısı bulunur.

Bunun yanı sıra dezavantajları ise:

- Değişken verinin işleniş açıısından zayıftır.
- Koordinat verisi eğiliminde sıkı olarak veri tabanı yönetimine bağlı değildir. Bu yüzden gezlilik ve bütünlük sorunu bulunmaktadır.
- Sorgulama ve modelleme için mekansal konum veya öznitelik verisine bağlıdır.
- Sorgulama ve modelleme CBS nin fonksiyonuna bağlı olarak sınırlı desteklenmektedir.

Nesneye dayalı veri tabanı CBS nin avantajları:

- Veri tabanındaki gerçek dünya objeleri ve bunların kavramları ile sunumları arasındaki semantik boşluk ilişkisel veri tabanına göre daha azdır.
- Nesneye ait özelliklerin ve fonksiyonların saklanması en minimum işlem gerektirir.
- Raster ve vektör yapıda veriler aynı veri tabanında toplanabilir.
- Objelerin veri alışverişi desteklenir.
- Özellikle karmaşık objelerin ve ilişkilerin sorgulanması daha az operasyona ihtiyaç duyulmasından dolayı daha hızlıdır.
- Daha az disk alanı talep edilmektedir. İlişkisel veri tabanı ise daha fazla indeks dosya saklama alanına ihtiyaç duyar.
- Kullanıcı tanımlı fonksiyonlar kullanılabilir değildir.

Dezavantajları ise:

- Nesneye dayalı veri modeli için uluslararası bir kabul görmüş standartlar bulunmamaktadır. Bu yüzden farklı veri tabanları farklı standartlarda üretilmektedirler.
- Nesnelerin tanımlanması oldukça zordur, özellikle bu zorluk devamlı mekansal objelerde ortaya çıkar.
- Fonksiyonlar ve topolojiyi tanımlanması obje kadar iyi olmalıdır.

### 3.7. CBS Amaçlı Sorgulama Dilinde Bulunması Gereken Özellikler

Geleneksel veri tabanı sistemleri coğrafi uygulamaların ihtiyaçlarını tam olarak karşılayamamaktadırlar. Temel veri modelleri altında sorgulama dilleri ve erişim

yöntemleri basit veri örnekleri ile tanımlanır. Buna örnek olarak tam sayılar ve karakterler verilebilir. CBS ler ise mekansal veriyi işler ve mekansal veri karmaşık ve semantik bir yapıya sahiptir. Geleneksel veri tabanı yönetim sistemleri coğrafi uygulamalarını ihtiyaçlarına cevap verebilecek biçimde etkili geometrik operasyonlar ile desteklenmez. Ayrıca geleneksel veri tabanı yönetim sistemlerindeki kullanıcı arayüz coğrafi verinin gösterilmesi için uygun değildir. CBS uygulamalarının biçimsel ve nitinsel bir yapısı vardır. Sonuçlar ya kartografik olarak ya da nitinsel olarak görselleştirilirler.

Sorgulama çalışmalarında 8 farklı veri seçimi ve bunun sonucunda bir sorgulama için istenilen talepler bulunur. Bunlar;

- Mekansal seçim kriteri: buna ilişkin örnek bir gölün 60 m çevresindeki Alisi kişi'nin sahibi olduğu parsellerin hepsini seçilebilir. Buna göre sorgulamanın ifadeleri “komşudur”, “ilişkilidir”, “içinde” veya “üzerinde” olabilir.
- Noktaya dayanarak seçim “Bu nedir?” “Bu kimin binasıdır?” gibi sorgulamalar örnek verilir ve değişkenler ekranda görülebilir.
- Sorgulama sonuçlarının birleşimi
- Mekansal içerik bir yerleşime ait kasabaların gösterilmesi ni istenmesi gibi.
- Sorgulama penceresinin seçimi.
- Öçeğin tanımlanması
- Harita işaretler tablosunun tanımlanması
- Öznitelik değerine dayanarak farklı sunular yapılmasıdır. [8]

Sorgulama dilleri için var olan, bir tanesinin genişletilmiş biçimi sorgulama işlemleri için kullanılabilir. Örnek olarak SQL verilebilir. Yapılacak istenilen çalışmaya uygun konuma getirilmek için sorgulama diline gerekli düzeltmeler ve eklemeler eklenebilir. Böyle bir tasarım karmaşık olabilir. Gerçekte CBS uygulamaları nesnelere ile bağlantılı geniş alanı kapsadığı için genel bir tanım olmadı gibi, kesin kabul edilmiş kurallarda bulunmamaktadır. Sorgulama dillerine yönelik pek çok çalışma bulunmaktadır.

Bir mekansal sorgulama dili temel olarak aşağıdaki fonksiyonlara sahiptir:

- Çok büyük sayıda geometrik olduğu kadar alfa nümerik değeri de içeren nesnenin yönetilmesi ve sorgulanması,
- Nesnelerin karmaşık yapılarının kendince sunulması,
- Bu nesnelere geometrik operasyonlara dayanarak uygulanabilir yapılar.

Açıkçası genişletilmiş ilişkisel sistemler, ilktalepleri karşılamak için oldukça iyi bir olanaktır. Genişletilmiş ilişkisel sorgulama dilleri genişletilmiş kullanılarak yapılan pek çok çalışma bulunmaktadır. SQL buna örnek olarak verilebilir.

CBS amaçlı bir sorgulama dilinde

- Mekanal veri türleri desteklenmelidir,
- Mekanal verilerin geometrik ve topolojik karakterli sorgulamalarına olanak verilmelidir.
- Mantıksal operasyonlar yardımıyla karmaşık sorgulamalar formüle edilebilmelidir.
- Sorgulamanın formüle edilmesi kolay ve anlaşılır biçimde gerçekleştirilebilmelidir.

Sorgulama işlemleri hem coğrafi analiz kapsamında, hem veri işleme kapsamında ve hem de veri sunma kapsamında yer alabilir. Bir CBS'nin uzun hesaplara gerek olmadan cevaplayabileceği soruların toplamı o CBS'nin "sorgulama uzayı" olarak adlandırılır. Bu uzayın gerek genişliği, gerekse cevap süre performansı büyük ölçüde verinin depolandığı mantıksal ve fiziksel coğrafi veri yapısına bağlıdır. Sorgulama uzayı sonuçta bir sorgu listesi dir. Bu sorgulama tipleri şu grupta toplanabilir.

- Grafikten veri tabanına sorgulamalar
- Veri tabanından grafiğe sorgulamalar
- Metrik sorgulamalar
- Topolojik sorgulamalar
- Düzen sorgulamaları
- Veri tabanı sorgulamaları [ 3, 9, 24, 25 ].

## 4 SUPPORT GIS YAZILI MI

Yukarıda tanımlanmakta olan nesneye dayalı veri modellemesine uygun yapıda geliştirilen Support GIS programı C++ ile geliştirilmiş ve programı destekleyebilecek içinde nesneye dayalı veri tabanı kullanılmıştır. Support GIS 1994 yılında Federal Almanya Cumhuriyeti Bonn Üniversitesi Kartografya ve Geoinformasyon Enstitüsü Jeodezi ve Fotogrametri Mühendisleri ve Bilgisayar Mühendislerinden oluşan bir grup tarafından geliştirilmiştir.

Support GIS programı ile aşağıdaki bazı çalışmalar amaçlanmıştır.

- Geometri ve öznitelik verileri aynı veri tabanında tutulmuş ve böylece veri tutarsızlığı sorunu giderilmeye çalışılmıştır.
- Aynı veri tabanlarında tutulmakta olan öznitelik ve geometri verileri sorgulamada bazı gecikme sorunlarına yol açmaktadır. Support GIS ile bu sorun da çözümlenmeye çalışılmıştır.
- Mekansal verilerin modellenmesinde ilişkisel modelden uzaklaşmak sorunu çözümlenmeye çalışılmıştır.
- Mekansal verilerde analiz esnasında yetersiz kalan SQL de yaşanan sorunların önlenmesi amaçlanmıştır.

Aynı zamanda Support GIS ile kullanıcı aktif olarak uyarılmakta ve desteklenmektedir. Böylece kullanıcı kendisini bir uzman ile çalışıyormuş gibi hissedebilmektedir [ 3 ].

### 4.1 Support GIS'in Temel Yapısı

Support GIS programı iki temel dayanmaktadır. Bunlardan birincisi şema kavramı ve ikincisi ise sorgulama kavramıdır. Burada her iki uygulamaya da burada değinilecektir. Şema kavramı içerisinde yapılan işlemler nesneye dayalı düşüncüyü destekleyici biçimde verilmiştir.

#### 4.1.1 Support GIS’ de kullanılan şema

Yapılmakta olan bir çalışmada için daha öncede belirtildiği gibi, bütün ayrıntıların düşünülmesi ve daha sonra herhangi bir şüpheye yer veremeyecek şekilde çalışmaların planlanması oldukça önemlidir. Support GIS programında da planlama başlangıcı olarak şema bölümü çok düzenli bir yaklaşımla tasarlanmalıdır. Nesneye dayalı veri modellemesinin getirmiş olduğu bazı özelliklerden dolayı birbirine bağımlı olarak gelişen bir mekanizma bulunmaktadır. Özellikle kullanım kolaylığı sağlayan ve tekrarı önleyen miras (inheritance) kavramı da burada işlenmektedir.

Support GIS programında şema oluşturmaya, öncelikle sınıf belirlenerek başlanır. Sınıflar belirlenirken bir sınıfın diğerine miras olarak bırakabileceği öznitelikleri olacaktır. Bu durumda bir üst sınıf belirlenir. Yeni oluşturulacak olan aktif sınıfa, seçilen üst sınıfın bütün öznitelikleri miras olarak kalır. Sınıflar oluşturulurken karmaşık objeler oluşturulabileceği gibi, aynı zamanda mekansal karakterli olan (örneğin mali) objeler de tanımlanabilir (Şekil EK B-1).

Sınıflara ait öznitelik verileri ise dört farklı şekilde oluşturulur (Şekil EK B-2).

- Objeye ve sınıflara ait “lokal öznitelikler”
- Değerleri başka bir obje sınıfı tarafından belirlenmiş “uzaktan alınan öznitelikler”
- İlişkisel SQL veri tabanından alınan “hariçi öznitelikler”
- Değerleri bir fonksiyon yardımı ile elde edilen “fonksiyonel öznitelikler” [ 3 ]

Support GIS de bulunan “tür (art)” kısmı sayesinde özniteliklerin cinsi belirlenebilir.

- Doğrudan (Direkt)
- Uzaktan (Remote)
- Tablodan (Tablo)
- Fonksiyon (Fonksiyon)

“Direkt” öznitelikler ile kullanılan indeks kısmı ise seçili objenin tek anlamı mı? yoksa çok anlamı mı? yorumlanacağı belirlenir. Aynı numaraya sahip olup oluncayaacağı “indeks” menüsü sayesinde sağlanır.

- Yok (İndeksi z)
- Tek Anla mı (E ndeuti g)
- Tek Anla mı De ği ŝtirilemez (E ndeuti g – İ nakti v)
- Çok Anla mı (M ěhrfach)
- Çok Anla mı De ği ŝtirilemez (M ěhrfach – İ nakti v)

Ayr ı ca her s ı nıfta bir defa kullanılabilen “tanıtıcı, karakteristik öznitelik (bezeichner attribut)” sayesinde ilgili objenin tanımlanmasında karakteristik rol oynar.

Uzaktan öznitelikler olarak adlandırılan “Remote” öznitelik ise daha önce tanımlanmış olan iki obje arasındaki ilişki sonucu özniteliğin alınmasını sağlar. Özniteliğ alınacak olan sınıf belirlenir ve aralarındaki ilişki seçilir.

Harici öznitelikler (Tablo) sayesinde daha önceden hazırlanmış olan ACCESS ve EXCEL dosyalarından doğrudan veri alınabilir. Bunun için bazı de ği ŝikliklerin yapılması gerekmektedir.

Fonksiyonel öznitelikler sayesinde öznitelik de ğerleri hesaplanabilir. E ğer daha önceden tanımlanmış bir de ğer var ise bu durumda bazı parametrelerin belirlenmesi gerekir.

Support GIS de bulunan di ğer bir kavram ise verinin “tip” özelliğidir. Tip menüsü ile seçilen özniteliğın

- Tam Sayı ( Ganzzahl)
- Ondalıklı Sayı ( Gleitzahl)
- Karakter ( Zeichenkette)
- Evet, Hayır gibi cevaplar ( Logisch)
- Tarih Tipli Veri ( Datum)

gibi veri tipi üstlenmesini sağlar. Yapılan çalışmaya ait bütün öznitelik girişleri ekte verilmiştir.

Sınıflar arasındaki ilişkilerin tanımlanması daha önceden tanımlanan “kardinallik” ve “semantik” özellikler ile sağlanır. Objelerin arasındaki ilişkiler;

- Mesleki (disiplinel)
- Mekanal olarak iki farklı bölüme ayrılır.



“M sleki (dişiplinel) iliřkiler”, (sahiptir, ait, kiracı) gibi tanımlanır.

“Mekansal iliřkiler” ise, kendi arasında iki farklı kısma ayrılır.

- a. Metrik
- b. Topolojik

Mekansal iliřkiler ya metrik ya da topolojik olarak kurulurlar. Metrik olarak kurulan iliřkiler (Yakın<20m) Yani 20 metreden yakın şekilde, topolojik olarak kurulan iliřkiler ise (içinde, üzerinde, kesişiyor, dışında..) gibi iliřkilerdir. İliřkilerde diđer bir özellik ise “kardinalliktir”. Kardinallik sayesinde bir sınıfın diđer sınıfla olan iliřkisi farklı bir biçimde tanımlanır. Buna göre bir parselde sadece bir tane binanın bulunur iliřkisi 1:1 ile kurulur. Bunun yanı sıra

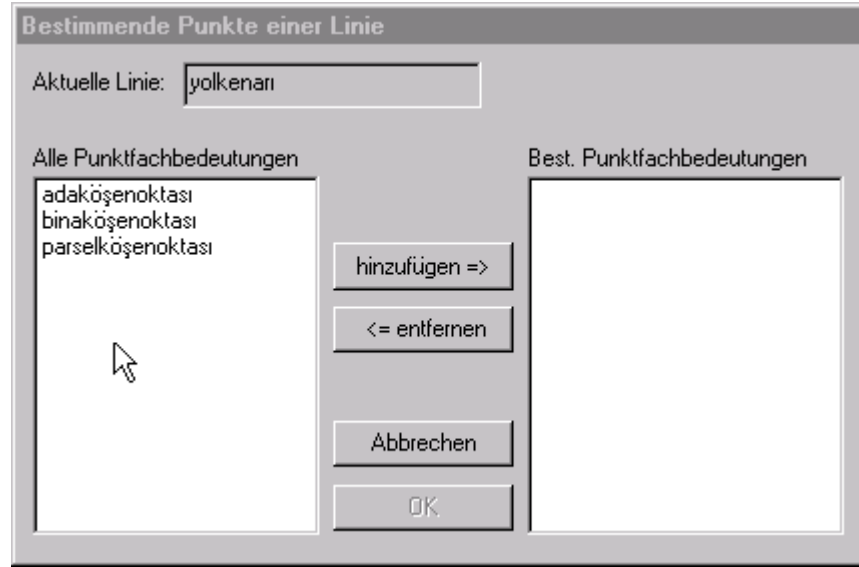
- 1:1
- 1:n
- n:1
- 0:0 kardinallikleri bulunmadır (Şekil EK B-3).

“0” sayısı sonsuzluđu işaret etmektedir.

Support GIS şemada bulunan diđer bir düşünce ise yersel referans elemanlarıdır (Şekil 4.1.a, Şekil 4.1.b). Mekansal referans elemanları alan, çizgi, nokta ve yazı (text) şeklindedir. Bu elemanların oluşturulması ise önce en alt kademeden başlanılarak yapılır. Buna göre en alt kademede bulunan, nokta elemanıdır. Çizime ait bütün nokta elemanları tanımlanır. Buna köşe noktası, parsel köşe noktası vb. buna örnek olarak verilebilir. Bu aşamada en alt kademeden başlamak, buna bađlı olarak tanımlanacak olan varsa çizgi ve alan objelerini tanımlamasına da yardımcı olacaktır



Şekil 4.1.a: Support GIS de çizgi objelerinin oluşturulması



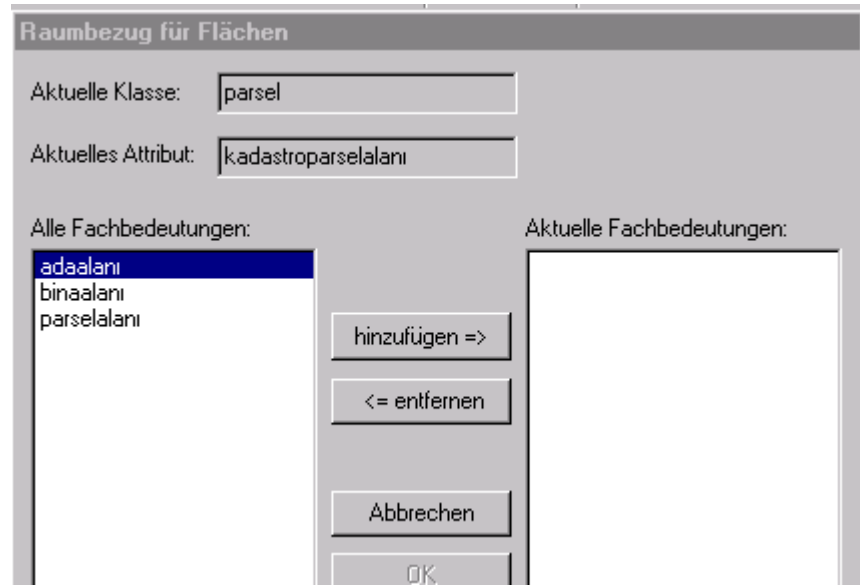
Şekil 4.1.b Çizgi objeleri oluşturulacak olan noktaların belirlenmesi

Çizgi objeler oluşturulurken yapılacak çizgi obje adlandırıldıktan sonra oluşturma safhasında bütün noktaların görüldüğü bir menü çıkmaktadır. Buradan ilgili çizgiyi oluşturabilecek bütün noktalar aktif hale getirilebilir. Böylece oluşturulan çizgi obje için gerekli işlemler tanımlanmış olur. Aynı özellikler alan objeler içinde geçerlidir. Fakat burada artık çizim objeleri kullanılır. Yazı elemanları (Şekil 4.2) ise ölçülmiş olan noktaları, hastane ve okul gibi isimler olarak belirlenir.



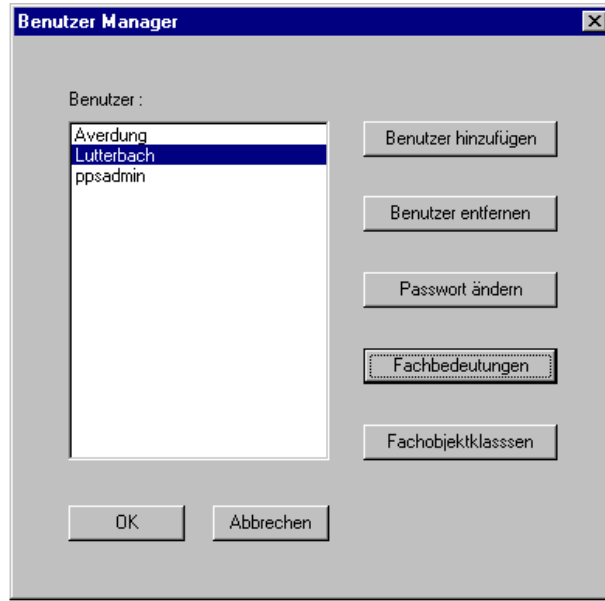
Şekil 4.2: Support GIS de yazı elemanlarının girilmesi

Çizim elemanları ve sınıflar arasındaki ilişkiler “mekansal referans (raumbezug)” yardımı ile kurulur. Çizim elemanları ile sınıflar arasındaki ilişki için (Şekil 4.3) tanımlanan sınıfa ait özneliğin ilgili çizim objesinde bulunması gerekir. Bir parsel için parsel alanı bir öznelik olarak sunulur. Bu eğer çok anlamlı olarak belirtilir ve fonksiyon öznelik olarak girilirse, özneliğin oluşturulması esnasında karşımıza ilgili bir menü gelecektir. Burada ilgili aktif sınıfın ismini yazacak ve bu özneliğin hangi çizim objelerinden oluşabileceği belirlenebilir. Aşağıda bu uygulamaya ait bir görüntü bulunmaktadır.



Şekil 4.3: Sınıflar ve çizim objeleri arasındaki ilişkilerin tanımlanması

Şe na böl ü münde bulunan di ğer bir ö ne mi kavram ise kullanıcı nı tanı ml anabil mesi ve böylece kullanıcı ların rastgele programa eriş i mi ni n sağ lanması dır (Ş ekil 4.4). “Kullanı myöneti mi (Benutzer management)” adı verilen bu böl ü mde her kullanıcı nın kendilerine ait ş ifreleri bulunmakta ve müdahale edebilecekleri böl ü ml er farklı ol maktadır. Bir kısım kullanıcı sadece ç izi mobjelerine müdahale edebilirken, bir kısım kullanıcı obje sınıflarına eriş ebilirnektedir. Böylece verilerde güvenlik ve tutarlılık sağ lanmaktadır.



Ş ekil 4.4: Support GIS de kullanıcı yöneti mi

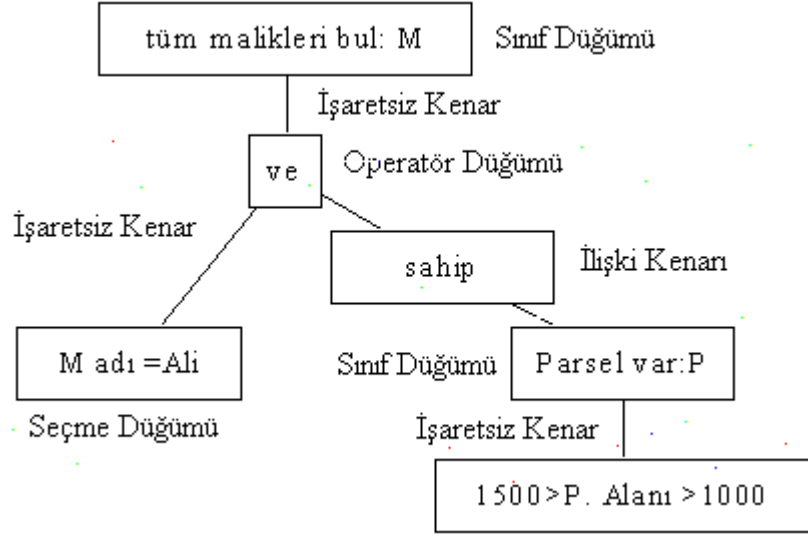
#### 4.1.2 Support GIS de kullanılan sorgulama

Sorgulama araçları sayesinde veri tabanına aktarılmış olan veriler amaca uygun bir biçimde tablolar, yazılı metinler veya kartografik olarak sunulur. Support GIS sorgulama diline bazı yetenekler eklenmiştir.

- Mekanal veri türlerine (nokta, çizgi, alan) ilişkin sorgulama ların yapılması
- Mekanal ilişkilerin sorgulanması na olanak tanınması
- Mantıksal operasyonlar kullanılarak (ve, veya, eşit, eşit değil) karmaşık sorgulama ların for müle edilmesi.

Sorgulama işle mi sırasında bazı grafikler kullanılmakta ve bu da kullanıcı ya bazı kolaylıklar sağ lamaktadır. Sorgulama işlemleri şekillerden ve yazılardan oluşur.

Şekiller düğümler ve kenarlar (Şekil 4.5) olarak adlandırılır. Buna ilişkin bir örnek aşağıda verilmiştir. Düğümler ve kenarlara ait açıklamalar daha sonra yapılacaktır.



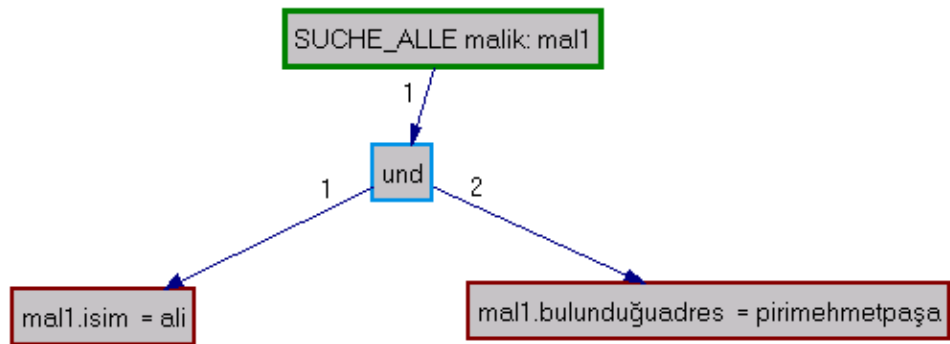
Şekil 4.5. Support GS de sorgulama işleminde düğüm ve kenarların gösterimi [3]

Sorgulama işlemleri üç farklı biçimde yapılabilir.

- Öznelik üzerinden sorgulama
- İlişkiler üzerinden sorgulama
- İlişkiler ve öznelikler ile yapılan sorgulama

#### Öznelik üzerinden sorgulama

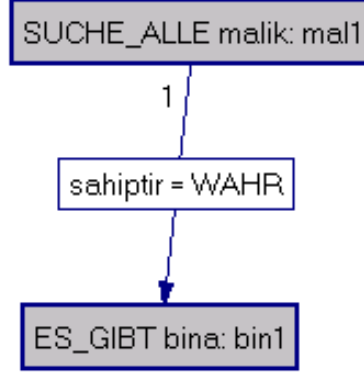
Burada sınıf ve sınıfa ait öznelikler birlikte sunumu oluştururlar (Şekil 4.6).



Şekil 4.6: Öznelikler kullanılarak yapılan sorgulama işlemi (malik adı ve adres üzerinden sorgulama)

## İlişkiler üzerinden sorgulama

Arada sınıflar arasında daha önceden tanımlanmış olan ilişkiler kullanılır. Buna ait sorgulama “Hangi malik bir eve sahiptir?” şeklinde olabilir. Buna ilişkin sorgulama grafiği ifade edilerek istenirse (Şekil 4.7);

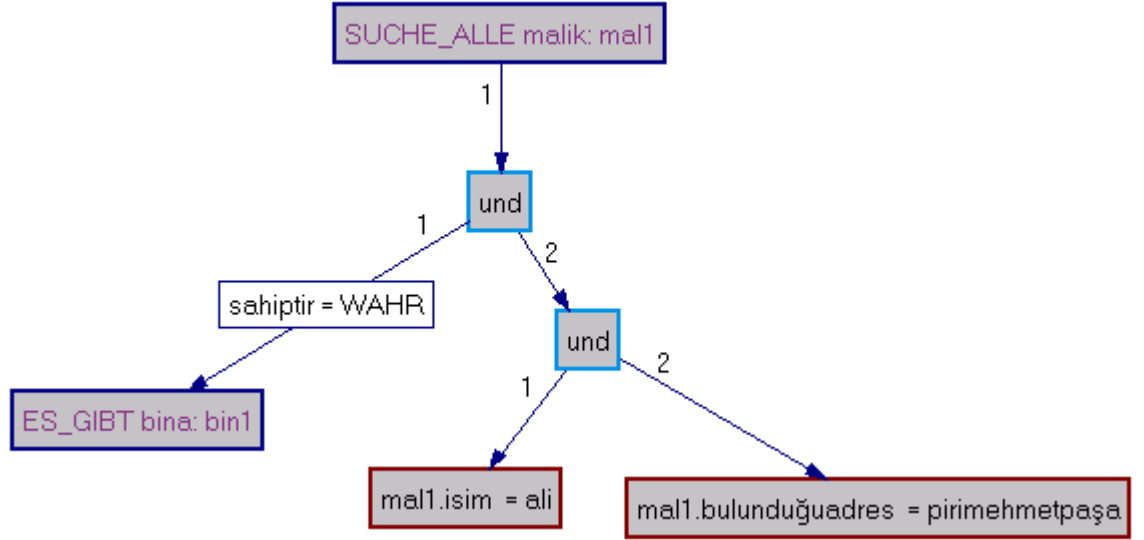


Şekil 4.7: Sınıflar arası ilişkilerin sorgulanması işlemi

## İlişkiler ve öznitelikler ile yapılan sunular

Burada hem daha önceden sınıflar arasında kurulmuş olan ilişkiler hem de öznitelikler ile karışık bir sorgulama oluşturulur (Şekil 4.8). Sorgulama “Adı Ali olan, Piri Mehmet paşa’da oturan ve bir binaya sahip olan kişileri bulunuz” şeklinde olmaktadır.

Burada sınıf olarak Malik ve Bina, Öznitelikler ise İsim ve Mahalledir. Sorgulama grafiği ise;



Şekil 4.8: Karar sorgulama işlemi

İlişkiler daha önce Support GIS şemasında belirtildiği gibi nesnelere (disiplinel) veya metrikte dâbilir.

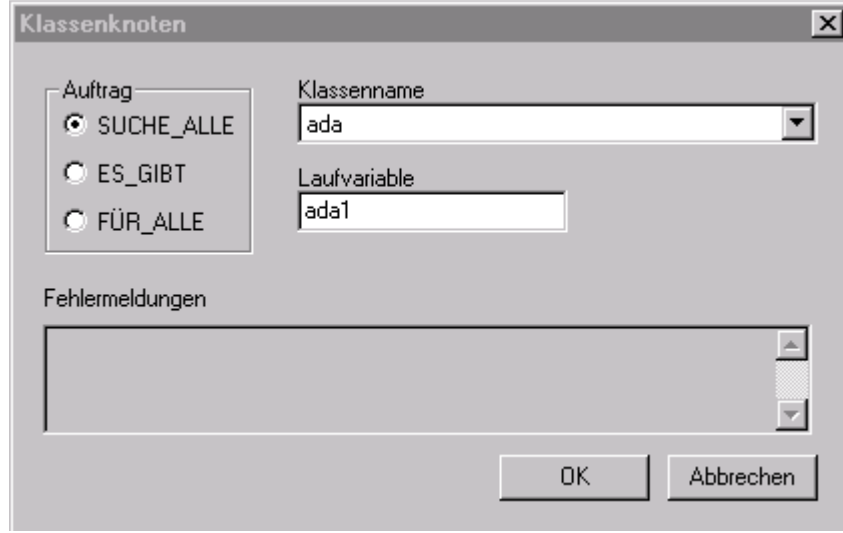
Sorgulama işleminde kullanılan düğümler işlenecek olursa:

- Sınıf Düğümü (Klassenknoten)
- Hesap Düğümü (Blanzierungsknoten)
- Operatör Düğümü (Operatorknoten)
- Rapor Düğümü (Reportknoten)
- Seçme Düğümü (Selektionstestknoten)
- Başlama Düğümü (Startknoten)

#### **Sınıf Düğümü (Klassenknoten)**

Bu menü sayesinde sınıf düğümleri üretilir. Sınıf düğümleri çizimobjeleri tarafından sunulur. Buna göre üç farklı sınıf düğümü oluşturulabilir (Şekil 4.9):

- SUCHE\_ALLE (belirli bir koşul veya koşulları yerine getiren tüm objeler bulunur)
- ES\_GIBT (en az bir obje için ilişki sorgulaması yerine gelmelidir).
- FÜR\_ALLE (Her obje için ilişki sorgulaması yerine gelmeli ve koşul tüm objeler için sağlanmalıdır.)



Şekil 4.9: Sınıf düğümlerinin oluşturulması

### Hesap Düğümü (Blanzierungsknoten)

Sınıflara ait istenilen sayı, ortalama değer vb. gibi işlemler sorgulamaya katılabilen için hesap düğümü seçeneği kullanılır.

- Sayı ( ANZAHL)
- Ortalama Değer ( Mittel ) ( AVG)
- Maksimum Değer ( Maximum ) ( MAX)
- Minimum Değer ( Minimum ) ( MIN)
- Toplam( Summe ) ( SUM)

Sayı düğümü kullanıldığı zaman sadece sınıf kullanılmakta, öznelik ise gözükmemektedir.

### Operatör Düğümü ( Operatörknoten):

Eğer sorgulama içinde koşul kullanılacak ise operatör düğümü kullanılır. Operatör düğümünde kullanılan şartlar

- ve ( und)
- veya ( oder )

Yukarıdaki tanımlamalardan “ve” koşulunda tüm koşulların olması istenilir. Veya koşulunda ise ileri sürülen koşullardan sadece birinin objeye ya da objelerde bulunması yeterli görülmektedir.



### Seçme Düğümü (Selektionstestknoten)

Seçim düğümüleri sayesinde öznelik değerleri karşılaştırılır (Şekil 4.10).



Şekil 4.10: Seçme düğümüleri oluşturulması

Yukarıda verilen Şekil 4.5 de  $1500 > P$ . Alan  $> 1000$  sorgulamasında seçme düğümü kullanılmıştır.

### Rapor Düğümü (Reportknoten)

Rapor düğümü sayesinde sorgulama görünür hale gelir. Fareni sağ tuşuna basıldığı zaman oluşan menüde rapor düğümü tıklandığı zaman seçilmiş olan rapor görülecektir.

### Başlama Düğümü (Startknoten)

Başlama düğümü sadece tüm seçeneği ile birlikte ve sınıfla birlikte kullanılır. Fareni sağ tuşuna basıldığı zaman başlama düğümü oluşturulur. Eğer bir başlama düğümü tanımlanmış ise bir uyarı notu veya ikaz sesi gelmektedir. Başlama düğümü çevresi yeşil olarak tanımlanır.

Sorgulamada kullanılmakta olan diğer bir özellik ise kenarlardır. Buna göre Support GS de kenarlar;

- İşaretsiz kenarlar
- İlişki kenarları olmak üzere iki çeşittir.

### **İşaretsiz kenarlar**

İşaretlenmemiş kenar olarak adlandırılan kenar sınıf düğümlerini ve seçme düğümlerini bağlar. Her bir kenarın bir numarası bulunur. Aynı ilişki göstermeyen kenarlar birbirine bağlanır.

### **İlişki kenarları**

Yalnızca sınıf düğümünden ve operatör düğümünden başlayan ilişkili kenarda metrik veya mekansal ilişkiler kullanılır. Bu kenar sadece sınıf ile son bulur.

Sorgulamaların sonuçlarının gösterilmesi “veri kazanım menüsü (erfassungdialog)” kısmında bulunan bazı komutlar yardımıyla gerçekleştirilir. Öncelikle çizimobjeleri gözükür ve daha sonra oluşturulan çizimobjeleri sınıfları ile birleştirilirler.

Sorgulamalar oluşturulduktan sonra objelere ait verilerin girilmesi aşamasına gelir. Bu işlem “büyük miktarlarda veri girişi (massenerfassung)” kısmında gerçekleştirilir (Şekil 4.11). Bu işlem ile birlikte verilere ait bilgiler girilir ve girilirken menü içerisinde bulunan ilişkiler kısmında hangi sınıf ile bağlantısı var ise tanımlanabilir. Bu kolayca şöyle açıklanabilir. “E” sınıfı örnek olarak seçilirse; daha önceden şema kısmında şahıs sınıfı ile ilişkisi tanımlanmış olan bina sınıfı sorgulama işlemi için burada tekrar birleştirilebilir. Aşağıda anlatılan ilişkin bir şekil bulunmaktadıdır.

Massenerfassung

Fachobjektklassen:  
 şahıs

Inst./Attr.	soyad	cinsiyeti	babaadı	doğumtarihi	isim
11	alkuşak	<default>	ali	15.12.1967	ekrem
12	sarı	<default>	bayram	18.05.1978	ismet
13	-	<default>	-	<default>	ist.bel.
14	tevfik	<default>	ömer	23.06.1954	mehmet
18	ceylan	<default>	hayrullah	01.01.1932	mustafa
19	<default>	<default>	<default>	<default>	<default>

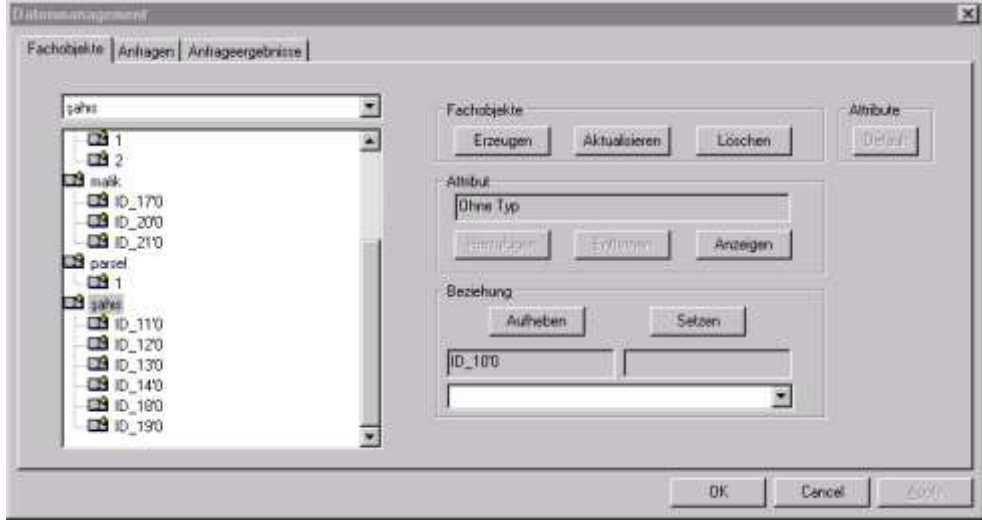
Beziehungen  
 şahıs%sahtiptir@bina%aittir   gesetzte Beziehungen anzeigen

Inst./Attr.	binano	binafonksiyonu	parselno
5	1	<default>	1
15	2	<default>	2
23	25	<default>	19

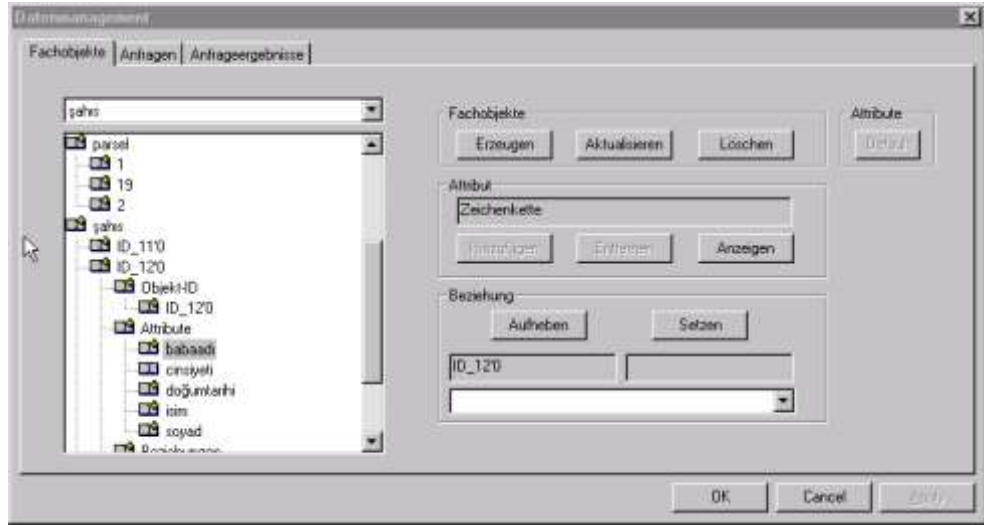
Şekil 4 11 Obje sınıfları ve şahıs sınıfı arası değerlerin oluşturulması

Ayrıca burada görülmekte olan kırmızı yazılı olan parsel numarası özelliğini bina sınıfı daha önceden belirtildiği gibi “remote” yani uzaktan almakta idi. Bunun sonucunda bina değerlerini girerken parsel değerine müdahale edilemez. Ancak parsel ve bina arasındaki ilişki eğer bu kısımda kurulur ise bu durumda “bina” sınıfı parsel numarası özneliğini otomatik olarak üstlenecektir. Bunun için;

İlgili parsel ile ilgili bina çift olarak aktifleştirilir. İkiisi arasında “setzen” adı verilen “kurulum” işlemi onaylanır. Bu durumda 19 numaralı parsel 25 numaralı bina ile ilişkilendirilmiş olur. Bunlara ilişkin öznelik üstlenmesi ise giriş işlemlerinin yapıldığı kısımda görülebilir (Şekil EK B-4).

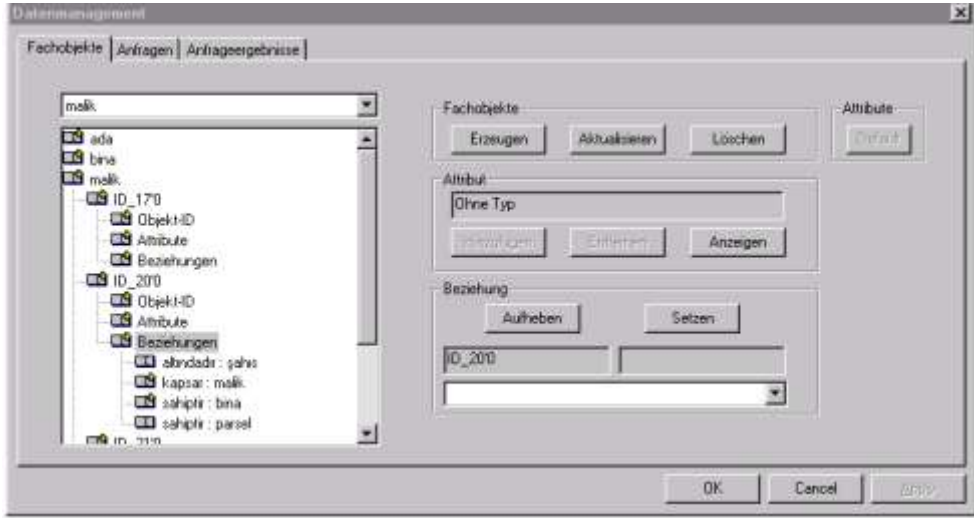


Şekil 4 12: Sınıflara ait oluşturulan ID numaraları



Şekil 4 13: Oluşturulan ID değerlerine karşılık gelen öznelik gösterimi

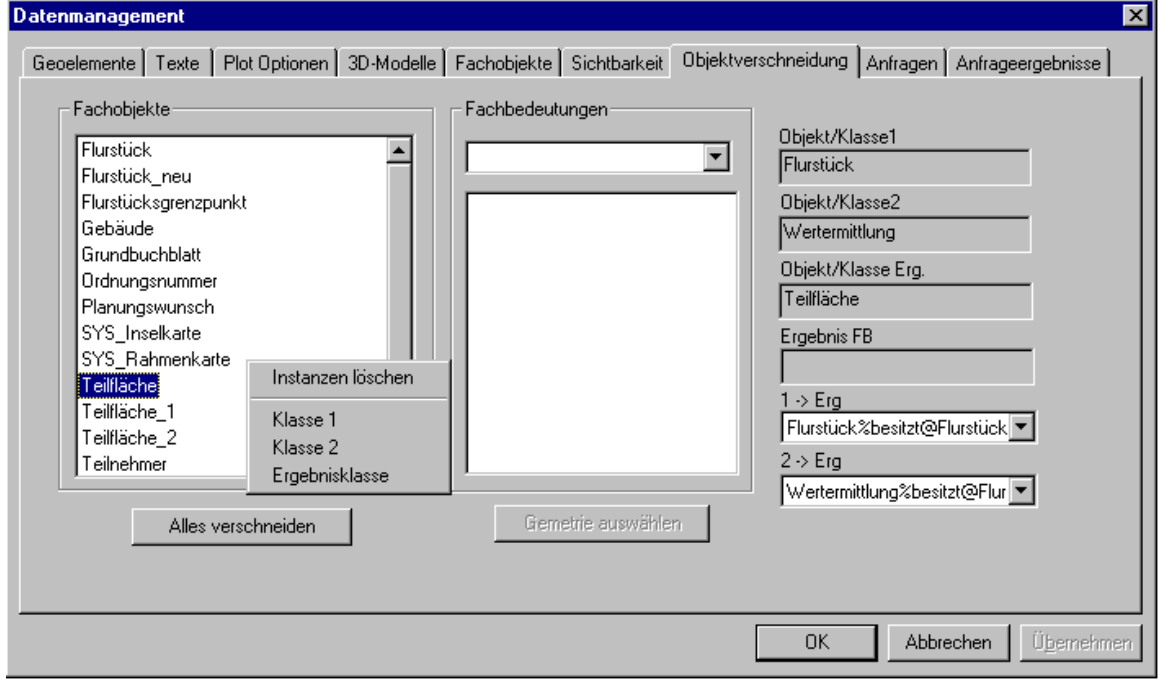
İlişkilerin oluşturulmasından sonra çizim objeleri belirli ID'lere sahip olacaklardır (Şekil 4 12, Şekil 4 13 ve Şekil 4 14).



Şekil 4.14: ID değerlerine karşılık gelen ilişkilerin gösterimi

#### 4.1.3 Support GIS’ de grafik işlemleri:

Support GIS de yapılacak olan grafik işlemleri için “**gösterim (anize)**” menüsü kullanılarak açılacak olursa burada bütün sınıflar çizim objeleri ile birlikte gösterilir. AutoCAD destekli olan Support GIS grafiği AutoCAD de bulunan he men hem en bütün menülere sahiptir. Layer özelliği sayesinde objelere ait çizim tipleri veya renklerde değişiklik yapılabilir veya istenilen layer gerektiği zaman dondurularak korunmaya alınabilir. Öncelikle “**giriş (erfassen)**” menüsü sayesinde ilgili nokta, çizgi, yay, alan objeleri ile çizim objeleri eşleştirilir. Öncelikle çizim objeleri tanılandıktan sonra gerekli çizim elemanlarında gerekli sayısalştırma işlemleri yapılır. Aynı zamanda yapısal fonksiyonlar sayesinde çizgi üzerinde nokta olup olmayacağı ve kesişim hesaplaması gibi hesaplamalar yapılabilir. Bütün sorgulama işlemleri ve gerekli öznetelik verilerinin değerlerinin girilmesiinden sonra gösterimin kartografik olarak sunulması işlemi için obje taratması işlemi aktifleştirilir. Şekil 4.6 ya ait sorgulama sonucu (Şekil EK B-5) de gösterilmektedir.



Şekil 4.15: Sor gul a ma işle mleri ni n taranması

Burada birinci kısım da bütün sınıflar çizim objeleri ile birlikte sıralanır. Öncelikle taranması istenilen birinci sınıf seçilir daha sonra taranması istenilen ikinci sınıf. Sonuç objesi ise iki sınıftan taranması sonucu oluşur ( Şekil 4.15). Sonuç ve birinci obje arasındaki ilişki ve yine aynı şekilde ikinci ve sonuç objesi arasındaki sonuç ilişkisi görülür. Ve hepsinin taranması istenilerek ilgili sorgulama işlemi duşturulur.

#### 4.2 Support GS' in Genel Özellikleri:

Support GS Windows NT için geliştirilen bir sistem olup kolaylıkla ağ kurulumuna izin vermektedir. Sistem içerisinde veri tabanını hazırlayabilmek için Object Store veri tabanı kullanılmıştır. Grafik kullanımı için AutoCAD tercih edilmiştir. C++ dili ile programlanması sayesinde esnek bir kullanım sağlamıştır. Aynı zamanda internet ortamından da erişim sağlanabilmektedir. Aynı zamanda diğer kamu kuruluşlarında kullanılmakta olan sistemler ile de veri alışverişi kolaylıkla yapılabilir.

### 4.3 Support GIS' de Uygulanan İşlemler

Hasköy Bölgesi İmar Uygulamasına (EK C) ait yapılan çalışmaların bir kısmı Support GIS yazılımında denenmiştir. Bu çalışmaya ait bilgiler aşağıda bulunmaktadır.

#### 4.3.1 Sınıflara ait özniteliklerin oluşturulması

- Ada
- Malik
- Parsel
- Şahıs
- Etna

**ADA Sınıfının Oluşturulması ve Özniteliklerin Kazandırılması İşlemi:** (Tablo EK A-1)

a. “Tür” özniteliğinin eklenmesi:

Ada no, ilçe, kadastro alanı ve mahalle öznitelikleri için doğrudan öznitelik kazandırılmıştır. Eğer tablo özniteliği kullanılırsa bu durumda öznitelikler bazı özelliklerini daha önceden hazırlanmış olan Excel veya Access dosyalarından özniteliklere ait değerleri alabilirler. Hesaplanan ada alanı ise fonksiyon olarak belirlenmiştir. Fonksiyon olarak belirlenirken buradan alan hesabını yapabilecek hesaplama seçilmiştir.

b. Tip belirlenmesi işlemi:

Ada no, mahalle ve ilçe öznitelikleri birer sayı olmaları nedeniyle karakter tipidir. Kadastro alanı olarak yer yüzeyi seçilmiştir. Çünkü bu alan sayısında hesaplanan ada alanı bulunabilecektir. Hesaplanan ada alanı hesaplama sonunda koordinatlarından gelecek olan virgül sonrası rakamları taşıyacağından ondalıklı sayı olarak işaretlenmesi gerekir.

c. İndeks belirlenmesi:

Eğer bu seçenek yok olarak seçilir ise indeks yok olarak düşünülür. İndeks sadece direkt öznitelikle kullanılır.

d. Karakteristik öznitelik

Diğer önemli bir öznitelik ise “bezeichner attribut” yani “karakteristik özniteliktir”. Ada objesi için “ada no” karakteristik öznitelik olarak atanmıştır. Böylece sadece bir adamın bir tane numarası olması sağlanır. Bu özellik bir sınıfta sadece bir kez aktifleştirilebilir ve diğerlerine de etki eder. Bu özneliğin seçilmesi sonucu art özniteliğin yanında %işareti görülür.

e. Çoklu Öznitelik (Morfaches-Attribut)

Bu öznitelik sayesinde çizim objeleri ve sınıflar arasında ilişki kurulabilir. Bu öznitelik tip kısmına “\*” işaretini verir.

### **BİNA Sınıfına İlişkin Öznitelik Verileri (Tablo EK A 2)**

a. “Tür” özneliğinin eklenmesi:

Bina fonksiyonu, bina no gibi öznitelikler doğrudan veri girişi ile sağlanmak üzere belirlenmiştir. Bunun yanında binanın bulunduğu parsel no ise uzaktan bir öznitelik olarak alınmaktadır. Yani önce parsel numarası tanımlanmakta ve bina bu numaraya göre özneliğini edinmektedir.

b. Tip belirlenmesi işlemi:

Her üç tip de sadece numara ve yazıdan ibaret olduğundan karakter olarak belirlenmesinde herhangi bir sakınca bulunmamaktadır.

c. İndeks belirlenmesi:

Bina fonksiyonu çoğu kez aynı özellik olarak belirlenebileceği için çok anlamlı olarak seçilebilir. Bunun yanında bina numarası ve parsel numarası için hiç bir şey belirlenmeyebilir.



d. Karakteristik öznitelik

Bina numarasının sınıf içerisinde sadece bir defa gözükmesi istendiği için daha doğrusu bir bina sadece bir numara taşıyacağı için karakteristik öznitelik bina no ya atanabilir.

e. Çoklu Öznitelik (Morfaches- Atribut)

**MALİ K Sınıfına At Öznitelik Verileri (Tablo EK A 3)**

a. “Tür” özniteliğinin eklenmesi:

Baba adı, bulunduğu adres, cinsiyet, doğum tarihi, isim ve soyadı gibi bilgiler tamamen şahıs sınıfının alt sınıfı nedeni ile tekrarlanmaya gerek duyulmadan doğrudan kazanılır. Bunlar tablo ile de kazanılabilen özniteliklerdir.

b. Tip belirlenmesi işlemi:

Baba adı, bulunduğu adres, isim soyadı, cinsiyet gibi öznitelikler yazı türünde olduklarından karakter özniteliği ile belirlenmelerine rağmen doğum tarihi özniteliği tarih özelliği taşımasından dolayı tarih olarak seçilmiştir.

c. İndeks belirlenmesi:

Herhangi bir indeks değeri seçilmiştir.

d. Karakteristik öznitelik

Herhangi bir karakteristik öznitelik seçilmiştir.

e. Çoklu Öznitelik (Morfaches- Atribut)

Hisse payı olarak çok anlamlı öznitelik belirlenmiştir. Aynı hisse payına sahip olan pek çok kişi ile karşılaştırılabilir.

**ŞAHİS Sınıfına At Öznitelik Verileri (Tablo EK A 4)**

a. “Tür” özniteliğinin eklenmesi:

Şahıs sınıfı, malik sınıfının üst sınıfı olduğundan dolayı bütün özniteliklerini malik sınıfına aktaracaktır. Buna göre baba adı, cinsiyet, doğum tarihi, isim ve soyadı gibi öznitelikler doğrudan verilir. Bu öznitelikler aynı zamanda bir excel veya access tablosundan da rahatlıkla alınabilir.

b. Tip belirlenmesi işlemi:

Baba adı, cinsiyet, isim ve soyadı gibi öznitelikler karakter özelliğine sahip olabilir.

c. İndeks belirlenmesi:

İsim ve soyadı öznitelikleri çok anlamlı olarak düşünülebilir. Bunun anlamı soyadı ve adın bir çalışmada çok kez ortaya çıkabileceğidir. Fakat bu özneliği aktif olmayı seçmek anlamlı olacaktır. Böylece değiştirilemez. Eğer aktif olmayan çoklu anlamselir ise “[ ]” işareti görülür.

d. Karakteristik öznitelik

Bu sınıfa ait herhangi bir karakteristik öznitelik seçilmiştir.

e. Çoklu Öznitelik (Mrfaches- Atribut)

Bu sınıfa ait çok anlamlı bir öznitelik seçilmiştir. Sınıfta çizgi, alan ve nokta objesi bulunmaktadır:

#### **PARSEL Sınıfına Ait Öznitelik Verileri (Tablo EK A 5)**

a. “Tür”özniteliğinin eklenmesi:

Parsel sınıfı, ada sınıfının alt sınıfı olarak seçildiği için bazı öznitelikleri doğrudan ada sınıfından gelmektedir. Bunun dışında parsele ait olan cilt no, edinme şekli, edinme tarihi, parsel fonksiyonu, parsel no ve yevmiye gibi öznitelikler direkt olarak verilmektedir. Hesaplanan parsel alanı ise fonksiyon olarak belirlenmektedir.

b. Tip belirlenmesi işlemi:

Parsel no, parsel fonksiyonu, yevmiye ve edinme şekli gibi bilgiler karakter olarak belirlenmiştir. Kadastro parsel alanı yer yüzeyi olarak belirlenmiştir. Bunun nedeni, daha sonra hesaplatılacak olan hesaplanan parsel alanı için bir parametre

oluřturabilmesini sađlamaktadır. Glt no tam sayı olarak sečilmiştir. Hesaplanan parsel alanı ise ondalıklı sayı olarak belirlenmiştir. Edinme tarihi ise tarih tipi verilerek tanımlanmıştır.

c. İndeks belirlenmesi:

Parsel no, çok anlamlı olarak belirlenmiştir. Çünkü bir parsel birden fazla kişiye ait olabileceđi için birden fazla görülebilir. Aynı şekilde yevmiye de birden fazla görülebileceđi için çok anlamlı olarak belirlenebilir.

d. Karakteristik öznelik

Parsel objeleri için karakteristik öznelik olarak parsel numarası sečilmiştir. Böylece bir parsel sadece bir numara almaktadır.

e. Çoklu Öznelik (Morfaches- Atribut)

Kadastro parsel alanı burada çok anlamlı olarak sečilmiştir. Bu sayede parsel sınıfı ve alan objesi arasındaki ilişkiler kurulabilmektedir.

## 17. SONUÇLAR ve ÖNERİLER

Günümüzde farklı amaçlar ve kullanım alanlarına göre, farklı özelliklerde hazırlanmış CBS ortaya çıkmıştır. CBS'nin hazırlanmasında kullanılan veri modellerinin hepsi de CBS için bazı önemli avantajlar sunmaktadır.

Geliştirilen veri modelleri arasında ilişkisel veri modelleri günümüzde çok yaygın olarak kullanılmaktadır. Özellikle ilişkisel veri modeli tek başına kullanılmayıp diğer veri modelleri ile birleştirilerek kullanıcıların hizmetine sunulmaktadır. Fakat ilişkisel veri modelinin mekansal verinin özelliği olan çok fazla bilgiyi içermesi ve karmaşık yapısı karşısında göstermiş olduğu bazı yetersizlikler sonucunda CBS üreticileri farklı arayışlar içerisine girmişlerdir.

Nesneye dayalı veri modeli kavramı bu noktadan itibaren göze çarpmaktadır. Özellikle insan düşüncesi biçiminde olan yapısı, karmaşık mekansal verinin kolaylıkla yönetilebilmesi, nesneye dayalı veri modelini bugün oldukça popüler hale getirmiştir.

Nesneye dayalı veri modelinin sahip olduğu miraslanma özelliği sayesinde objelere ait hiyerarşik düzen kolaylıkla kurulmakta ve tekrar önlenmektedir. Ayrıca, objeye ait güncellemelerin yapılmasında, sadece üst sınıfta yapılacak olan değişiklikler alt sınıfa da yansımaktadır. Böylece zaman kaybı, veri tutarsızlığı ve veri tekrarı işlemleri en aza indirgenmektedir.

Veri depolama özelliği sayesinde, kullanıcının veriye ulaşım belirli ölçüde korumaya alınmıştır. Böylece kullanıcının programlamaya ait bilgileri bilmesine gerek kalmamıştır. Ayrıca obje sınıfları kolaylıkla oluşturulabilmekte ve sınıflar arasındaki ilişkiler tanımlanmaktadır. Nesneye dayalı kavramlar, ilişkisel veri modellerinden farklı olarak dinamik bir biçimde tutulmaktadır. Bu özellik ise operasyonlar ile sunulmaktadır. Objeler arasında mesaj geçişi sayesinde objelerin kendilerine ait işlemleri dinamik bir biçimde yapabilmeleri sağlanmıştır.

Özellikle nesneye dayalı özellikler kullanılarak geliştirilen Support GIS yazılımında da, bu özellikler göze çarpmaktadır. Sınıflar kolaylıkla oluşturulmakta, sınıflar arasındaki ilişkiler tanımlanabilmekte ve miraslanma özelliği objelere yüklenebilmektedir.

Programın şema kısmının düzenli bir biçimde oluşturulması sonucunda sorgulamada sırasında karşılaşılabilecek sorunlar ortadan kalkmaktadır. Özellikle sorgulamanın grafik ile desteklenmesi sayesinde kullanım kolaylığı sağlanmıştır. Sorgulamada topografik ve disiplineller kriterler kullanılabilirlerdir.

Ayrıca EXCEL ve ACCESS gibi sistemlere olan uyumluluğu sayesinde Support GIS yazılımında veri girişi kolaylıkla yapılabilirlerdir. Support GIS in sahip olduğu grafik gösterimi için çok yaygın olarak kullanılan AUTOCAD programının tercih edilmesi kullanıcıya getirilmiş olan diğer bir kolaylık olarak göze çarpmaktadır.

Support GIS yazılımının internette sunumu ile birlikte daha geniş kitlelere ulaşması kaçınılmazdır. Özellikle JAVA ve C++ programları desteği sayesinde CBS için gerekli duyulan bir sistemin gerçekleştirilmesi mümkün olmuştur. Günümüzde nesneye dayalı veri modeli kullanılarak yapılan çalışmalar hala devam etmektedir.

## KAYNAKLAR

- [1] **Ucuzal, L.**, 1999. Merkezi ve Mahalli İdareler’de Bilgi Sistemleri ve Uygulama Geliştirme üzerine Bir Tartışma, *Yerel Yönetimlerde Kent Bilgi Sistemi Uygulamaları Sempozyumu*, Trabzon
- [2] **Medeiros, C B and Hres, E.**, 1999. Database For Gs, *Seminar “Verwaltung Geographischer Daten in DBMS”*, DCC-Imecc- Unicamp, Brazil
- [3] **Uçar, D, Mergenstern, D, ve Cristoph, A.**, 1999. Nesneye Dayalı CBS Kavram ve Support GS Yazılımı s.41-54, *Harita Dergisi*, Ankara
- [4] **Kraak, M.J. and Ormeling, F.J.**, 1996. Cartography: Visualization of Spatial Data, Longman, England
- [5] **Batuk, E.G.**, 1997. Coğrafi Bilgi Sistemi Dersi, Arazi Bilgisi Ders Notları, Y. T. Ü, İstanbul
- [6] **Procopiu, O.**, 1997. Data Structures For Spatial Database Systems, Spring 97, Term Duke University, North California, Durham USA
- [7] **Uğtekin, N.**, 2001. Coğrafi Bilgi Sistemlerinde Görselleştirme Yüksek Lisans Dersi, Basılmamış Ders notları, İ. T. Ü, İstanbul
- [8] **Hohl, P.**, 1998. Cartographic Visualitaton in GIS, GIS Data Conversion, 1998 On Word Press
- [9] **Batuk, G, Küllür, S., Sarbanoğlu, H, ve Toz, G.**, 1996. Veriden Bilgiye: Coğrafi Bilgi Sistemleri CBS’96 s. 34-47 *Coğrafi Bilgi Sistemleri Sempozyumu 26-28 Eylül 1996*, İstanbul
- [10] **Özkaya, O.**, 1998. Yaşadığımız Kent dünyaya Şımalı, *Tapu ve Kadastro Mifettileri Derneği Yayını “Milkiyet”*, 30, Ankara
- [11] **Chung, M, Cobb, M, Shaw, K and Arctur, D.**, 1995. An Object-Oriented Approach For Handling Topology In Vpf Products, *GIS/LIS 95 Proceedings*, Vol 1, 14-16 Nov 95. Naval Research Laboratory USA

- [12] **Rendén, A.**, 1997. Conceptual Modeling And Spatiotemporal Information Systems: How to Model Real World, “*ScanGIS’97 The 6<sup>th</sup> Scandinavian Research Conference on Geographical Information Systems*”, Stockholm Sweden, June 1-3
- [13] **Worboys M. F.** :1995. GIS A Computing Perspective, Taylor & Francis, London
- [14] **Hardy, P. G And Woodsford P. A.**, 1997. Mapping With Live Features: Object Oriented Representations, *ICC Conference*, Stockholm July 1997
- [15] **Tüzün T.**, 1990, Nesne temelli C++ Programlama Dilini Kullanarak Problem Çözümüne Bir Örnek, *Yüksek Lisans Tezi*, İ.T.Ü Fen Bilimleri Enstitüsü, İstanbul
- [16] **Elner E.** 1993. Basic Principles And Concepts Of Object-Oriented, November, Vienna
- [17] **Yearsley, C, Worboys, M, Story, B, Jayawardana, P. And Bofakos, P.**, 1994  
Computational Support for Spatial Information Systems: Models and Algorithms, Innovations in GIS 1, Taylor & Francis, London
- [18] **Paton, N, Cooper, R, Williams H and Trinder, P.**, 1996, Database Programming Languages: Deductive, Functional, Persistent and Object oriented Approaches, Prentice Hall/ Harvester, London
- [19] **Schildt, H** 2000. C++ Temel Öğrenim Klavuzu, Türkçe Basım Editörü: Dr. Cahit Akın, Alfa yayınları, Mayıs 2000, İstanbul
- [20] **Sarközy F.** 1999, GIS Functions 1 Technical University Budapest Department of Surveying
- [21] **Burrough, Peter A.; McDonnell, Rachael A.**: 1998. Principles of Geographic Information Systems, Oxford University Press, Oxford
- [22] **Healey R. G.**, 1992. Database Management Systems, Geographical Information Systems Principles and Applications, Longman scientific and Technical, New York

- [23] **Helokunnas, T**,1994. Object-Oriented Geographic Data Management, *EGIS/MARI'94 Fifth European and Exhibition on Geographical Information Systems, Article Citations Conference Proceedings*, Finland
- [24] **Scholl, M and Voissard, A**, 1993. Object-Oriented Systems for Geographic Applications an Experiment With O2
- [25] **Frank A, U And Mark, D, M** 1992. Spatial Query Languages And Database Interfaces, *Geographical Informations Systems Principles and Applications*, Longman scientific and Technical, New York



Tablo A-1 Ada Sınıfına Ait Öznitelik Verileri

Ada	Tür	Veri tipi	İndeks	Tamıncı-öznitelik	Çoklu-öznitelik
Ada.no	Direkt	Karakter	Yok	Ada.no	
Hesaplanan-ada alanı	Fonksiyon	Tam sayı	Yok		
İlçe	Direkt	Karakter	Yok		
Kadastro-adaalanı	Direkt	Geo-Flache	Yok		İşaretili
Mahalle	Direkt	Karakter	Yok		

Tablo A-2 Bina sınıfına ait Öznitelik özellikleri Tablosu

Bina	Tür	Veri tipi	İndeks	Tamamcı-öznitelik	Çoklu-öznitelik
Bina-fonksiyonu	Direkt	Karakter	çoklu		
Bina-no	Direkt	Karakter	Yok	Bina-no	
Parsel-no	uzaktan	Karakter	Yok		Çoklu-öznitelik

Tablo A 3 Malik sınıfına ait öznitelik özellikleri tablosu

Malik	Tür	Veri tipi	İndeks	Tamırcı-öznitelik	Çoklu-öznitelik
Baba adı	Tablo	karakter			
Bulunduğu adres	direkt	karakter			
Cinsiyet	direkt	karakter			
Doğum tarihi		Tarih			
Hisse payı	direkt	karakter			Çoklu-öznitelik
İsim	direkt	karakter			
Soyadı	direkt	karakter			

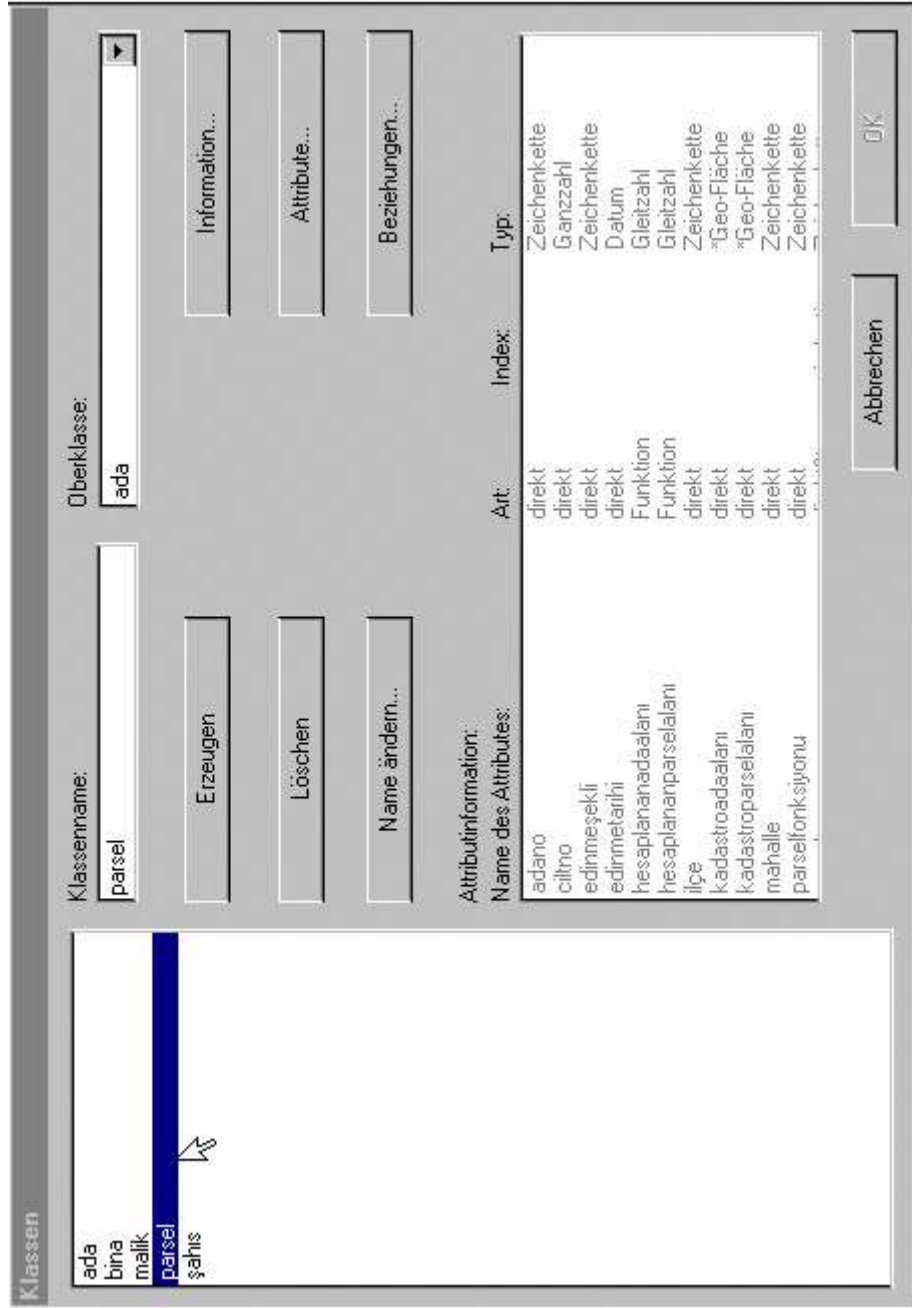
Tablo A-4 Şahıs sınıfına ait öznelilik verileri

Şahıs	Tür	Veri tipi	İndeks	Tanımlı-öznelilik	Çoklu-öznelilik
Baba adı	direkt	karakter			
Cinsiyet	direkt	karakter			
Doğum tarihi	direkt	tarih			
İsim	direkt	karakter	Çoklu-inaktif		
Soyadı	direkt	karakter	Çoklu-inaktif		

Tablo A 5 Parsel sınıfına ait öz niteliklerin gösterimi

Parsel	Tür	Veri Tipi	İndeks	Tamıtcı-Öz nitelik	Çoklu-Öz nitelik
Ada no	Direkt	karakter			
Cilt no	Direkt	Tam sayı			
Edinme şekli	Direkt	karakter			
Edinme tarihi	direkt	tarih			
Hesaplar aadaalanı	fonksiyon	ondalık			
Hesaplanan parsel alanı	fonksiyon	ondalık			
İlçe	direkt	karakter			
Kadastro adaalanı	direkt	Yer alanı			çoklu
Kadastroparsel alanı	direkt	Yer alanı			çoklu
Mahalle	direkt	karakter			
Parsel fonksiyonu	direkt	karakter			
Parsel no	direkt	karakter	Çok anlamlı	Tamıtcı-öz nitelik	
Yevmiye	direkt	karakter	Çoklu-inaktif		

## EK B



Şekil B-1 Support GIS de sınıf gösterimi

**Attribute**

Aktuelle Klasse:    Direkt  Remote  Tabelle  Funktion

Attributname:

Typ:   mehrfaches Attribut

Index:

Attributinformation:

Name des Attributes:	Art:	Index:	Typ:
adano	direkt		Zeichenkette
ciltno	direkt		Ganzzahl
edinmeşekli	direkt		Zeichenkette
edinmetarihi	direkt		Datum
hesaplananadaalanı	Funktion		Gleitzahl
hesaplananparselalanı	Funktion		Gleitzahl
ilçe	direkt		Zeichenkette
kadaastroadaalanı	direkt		*Geo-Fläche
kadastroparselalanı	direkt		*Geo-Fläche
mahalle	direkt		Zeichenkette
parselfonksiyonu	direkt		Zeichenkette
parselno	direkt%	eindeutig	Zeichenkette
yevmiye	direkt	[eindeutig]	Zeichenkette

Şekil B-2 Support GIS de öznitelik verilerinin oluşturulması

**fachliche Beziehungen**

Klasse 1:  Klasse 2:

Name 1->2:  Name 2->1:

Einstellungen

Kardinalität (0 = unbegrenzt)

1:1    m:1

1:n    n:m

Beziehungen der Klasse:

Klasse 2	Name 1->2	Name 2->1
parsel	üstünderdir	altındadır
şahıs	altır	sahiptir

vererbte Beziehungen:

Klasse 1	Klasse 2	Name 1->2	Name 2->1
----------	----------	-----------	-----------

Şekil B-3 Support GIS de sınıflar arasındaki ilişkilerin kurulması



Massenerfassung

Fachobjekt/lassen: bina

Neu Aktualisieren Übernehmen Abbrechen

Inst./Attr.	binano	binafonksiyonu	parselno
5	1	<default>	1
15	2	<default>	2
23	25	<default>	19

Beziehungen

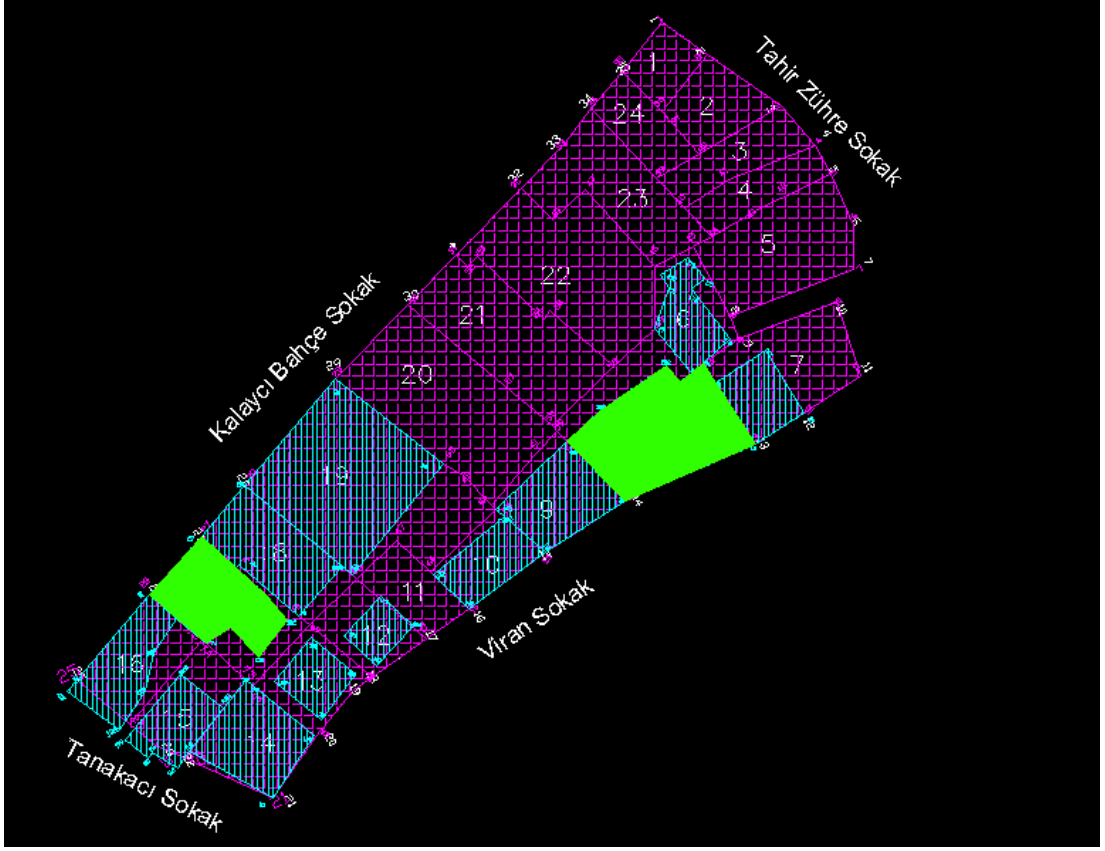
bina%üstündedir@parsel%altındadır

gesetzte Beziehungen anzeigen

parsel

Inst./Attr.	adano	ilçe	mahalle	hesaplanamadaları	parselno	parselfonksiyonu	edinmetarifi	yevmiye	edinmeşekli	hes
6	3016		primemelpaşa	0.000000	1	<default>	01.02.1999	350	Tapu Kutluğu	
22	<default>	<default>	<default>	0.000000	2	<default>	<default>	<default>	<default>	
24	3016	<default>	<default>	0.000000	19	<default>	<default>	<default>	<default>	

Şekil B-4: Bina ve parsel ait verilerin dışıdırılması



Şekil B-5 Sorgulama sonucu

## EK C

Tablo C 1 Hasköy Bölgesi ne ait malik bilgileri.

ad	soyad	doğum tarihi	baba ad	cinsiyeti	hisseyi
İST BLD			-	-	13/1
MEHMET	TEVFİK		OMER	E	6/48
MUSTAFA	CEYLAN		HAYRULLAH	E	14/1
MUSTAFA	CEYLAN		HAYRULLAH	E	13/48
İST. BEL			-	-	4/1
NURCAN	BASER		KASIM	K	20/1
MAHMUT	ERASLAN		İBRAHİM	E	9/2
MELAHAT	ERASLAN		HASAN	K	9/21
İSHAK	EKİNCİ		SERİF	E	21/1
NECATİ	PIRLANTA		MUHAMMET	E	13/2
SELAHATTİN	PIRLANTA		MUHAMMET	E	13/47
HATEM	KOÇ		İSMAİL HAKKI	E	1/1
KILMEN	BOZKURT		MUSTAFA	E	7/80
KILMEN	BOZKURT		MUSTAFA	E	22/80
NURHAYAT	TUZUN		MUSTAFA	K	2/80
NURHAYAT	TUZUN		MUSTAFA	K	7/80
ŞAKİR	BATALZEYNEP		TUZUNKİ	E	1/80
ŞAKİR	BATALZEYNEP		TUZUNKİ	E	5/80
ALİ	GÜRSOY	18.12.1976	İSMAİL	E	1/1
YILMAZ OTOMOTİV			-	-	1/1
BİRGÜL	AY		MEHMET	K	5/23
HİZİR	GUCLU		AHMET	E	5/23
HUSNİYE	ERENLER		SULEYMAN	K	2/23
NEBİYE	YAZGAN		İBRAHİM	K	2/23
SUNA	FAYDALI		TAHSİN	K	3/23
VURAL	DİZMAN		YUSUF	E	4/23
ZEKİ	DEMİRÇİ		HASAN	E	2/23
İST. BEL			-	-	1/1
İST. BEL			-	-	1/1
İST. BEL			-	-	11/1
İST. BEL			-	-	37/1
HALİLİBRAHİM	SOFU		DURSUN	E	1/8
MELAHAT	SOFU		HASAN TAHSİN	K	1/8
NURAY	EFENDİOĞLU		HASANTAHSİN	K	4/8
SALİH	GELİSLİ		DAVUT	E	2/8
İST. BEL			-	E	1/1
HAYİM	SALAMON		LAVBA	E	2/7
NESEM	SALAMON		LAVBA	E	2/7
SULTANA	SALAMON		LAVBA	K	1/7
YASEF	SALAMON		LAVBA	E	2/7
ALİ	CAY		HASAN	E	3/32
MUSA	DOĞAN		MUSTAFA	E	15/32
NESEM	TOPAFEL		BENŞİYON	E	6/32
SULTANA	TORAFEL		BENŞİYON	K	8/32
GÜL	BOTANK		-	-	1/24
AHMET	SEN		MEHMET	E	3/102
ALİ	ERACAR		İBRAHİM	E	3/102
ALİ	YAKIÇI		HASAN	E	3/102
AYSE	DOĞAN		BUNYAMİN	K	3/102
AZİZ	KABAĞCI		HUSEYİN	E	3/102

Tablo C 2 Hasköy Bölgesi kişilere ait parsel cilt no, yevniye, edinme şekli ve tarihi bilgileri.

<b>cilt no</b>	<b>edinme şekli</b>	<b>edinme tarihi</b>	<b>yevniye</b>
41/4071	TAPU KUT.	01/02/1999	350
41/4071	TAPU KUT.	01/02/1999	350
41/4072	TAPU KUT.	01/02/1999	350
41/4072	TAPU KUT.	01/02/1999	350
41/4072	TAPU KUT.	01/02/1999	350
41/4072	TAPU KUT.	01/02/1999	350
41/4073	TAPU KUT.	01/02/1999	350
41/4073	TAPU KUT.	01/02/1999	350
41/4074	TAPU KUT.	01/02/1999	350
41/4074	TAPU KUT.	01/02/1999	350
41/4075	TAPU KUT.	01/02/1999	350
41/4075	TAPU KUT.	01/02/1999	350
41/4075	TAPU KUT.	01/02/1999	350
41/4076	TAPU KUT.	01/02/1999	350
41/4077	TAPU KUT.	01/02/1999	350
41/4077	TAPU KUT.	01/02/1999	350
41/4077	TAPU KUT.	01/02/1999	350
41/4077	TAPU KUT.	01/02/1999	350
41/4077	TAPU KUT.	01/02/1999	350
41/4078	TAPU KUT.	01/02/1999	350
41/4079	TAPU KUT.	01/02/1999	350
41/4080	TAPU KUT.	01/02/1999	350
41/4080	TAPU KUT.	01/02/1999	350
41/4080	TAPU KUT.	01/02/1999	350
41/4080	TAPU KUT.	01/02/1999	350
41/4080	TAPU KUT.	01/02/1999	350
41/4080	TAPU KUT.	01/02/1999	350
41/4080	TAPU KUT.	01/02/1999	350
41/4081	TAPU KUT.	01/02/1999	350
41/4082	TAPU KUT.	01/02/1999	350
41/4083	TAPU KUT.	01/02/1999	350
41/4083	TAPU KUT.	01/02/1999	350
41/4084	TAPU KUT.	01/02/1999	350
41/4084	TAPU KUT.	01/02/1999	350
41/4084	TAPU KUT.	01/02/1999	350
41/4084	TAPU KUT.	01/02/1999	350
41/4085	TAPU KUT.	01/02/1999	350

## ÖZGEÇMİŞ

15.12.1976 da Çayeli/ RİZE de doğdu. İlk ve orta öğrenimini Serik/ Antalya’da lise öğrenimini Sungurlu/ Çorum’da tamamladı. 1994 yılında Yıldız Teknik Üniversitesi Jeodezi ve Fotogrametri Mühendisliği Bölümünü kazandı. 1999-2000 yılları arasında İstanbul Teknik Üniversitesi İngilizce hazırlık bölümünde eğitim gördü. Daha sonra İ. T. Ü. Jeodezi ve Fotogrametri Mühendisliğinde lisans üstü eğitime başladı.