

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK BİLGİSAYAR ANA BİLİM DALI
BİLGİSAYAR AĞLARI VE İNTERNET TEKNOLOJİLERİ
BİLİM DALI

NESNEYE YÖNELİK VERİ TABANI TASARIMI
ve
BİR UYGULAMA
(YÜKSEK LİSANS TEZİ)

Tezi Hazırlayan;
Hasan ERDEMİR
050861011

Danışman;
Dr. Yalçın ÖZKAN

İSTANBUL, 2008

YEMİN METNİ

Sunduğum Yüksek Lisans Tezimi, Akademik Etik İlkelerine bağlı kalarak, hiç kimseden akademik ilkelere aykırı bir yardım almaksızın bizzat kendimin hazırladığına and içerim. 17.01.2008

Hasan ERDEMİR

NESNEYE YÖNELİK VERİ TABANI TASARIMI

ve

BİR UYGULAMA

Tezi Hazırlayan: Hasan ERDEMİR

Özet

“Nesneye Yönelik Veri Tabanı Tasarımı ve Bir Uygulama” isimli çalışmamızda Nesneye yönelik veri tabanı ve özellikleri tanıtılmaya çalışılmıştır. Çalışma konumuzun temelini oluşturan veri tabanı olarak, açık kaynak kodlarına sahip db4Objects seçilmiş, Microsoft .Net ortamında, nesne yönelimli programlama dili olan C# programlama dili kullanılarak uygulama geliştirilmiştir. Uygulama konusu olarak Türk lojistik sektöründe önemli bir yere sahip olan Genel Antrepo Yönetimi ele alınmıştır.

Anahtar Kelimeler: Nesneye Yönelik Veri Tabanı, Genel Antrepo, Depo, Lojistik

OBJECT ORIENTED DATABASE DESIGN

and

AN APPLICATION

Presented by: Hasan ERDEMİR

Abstract

In “Object Oriented Database Design and An Application” , it is aimed to introduce object oriented database and its characteristics. Db4Objects which has an open source platform is chosen as a base for our application. The application is developed with one of the the object oriented programming languages called C# on the Microsoft .Net enviroment. Common Bonded Warehouse Management is chosen as the implemenentation topic since it is very important in Turkish logistics sector.

Key Words: Object Oriented Database, Common Bonded Warehouse, Warehouse, Logistic

İÇİNDEKİLER

Yemin Metni	
Juri Sayfası	
Türkçe Özet ve Anahtar Kelimeler	I
İngilizce Özet ve Anahtar Kelimeler (Abstract)	II
Şekiller Listesi	V
Tablolar Listesi	VII
Kısaltmalar	VIII
Giriş	

I. BÖLÜM

VERİ TABANI SİSTEMLERİNDE TEMEL KAVRAMLAR

1.1.	Veri ve Bilgi	1
1.2.	Dosya, Kayıt ve Alan	2
1.3.	Veri Yönetimi	3
1.4.	Veri Tabanı	4
1.5.	Veri Tabanı Yönetim Sistemi	5
1.6.	Veri Modeli	6
1.7.	Veri Tabanı Düzeyleri	7
1.7.1.	Dış Düzey	7
1.7.2.	Kavramsal Düzey	8
1.7.3.	İç Düzey	8
1.7.4.	Fiziksel Düzey	8
1.8.	Veri Modelleri	9
1.8.1.	Basit Veri Modelleri	9
1.8.2.	Modern Veri Modelleri	11

II. BÖLÜM

NESNEYE YÖNELİK VERİ TABANI

2.1.	İlişkisel Veri Tabanı ve Nesne Yönelimli Veri Tabanı	20
------	--	----

2.2.	Nesneye Yönelik Yaklaşım	21
2.3.	Nesne Yönelimli Veri Modelinin Özellikleri	22
2.3.1.	Nesneye Yönelimli Veri Tabanlarının Gelişimi	26
2.3.2.	NYVTS'nin Avantajları ve Dezavantajları	27
2.4.	Nesne Yönelimli Programlama Dilinin Özellikleri	28
2.5.	Nesne Yönelimli Modelleme Dili	31

III. BÖLÜM

NESNEYE YÖNELİK VERİ TABANI UYGULAMASI

3.1.	Sistem Analizi	42
3.1.1.	Gümrük	42
3.1.2.	Eşya İthalatı	43
3.1.3.	Taşıtların Türkiye Gümrük Bölgesine Giriş ve Çıkışı	44
3.1.4.	Eşyanın Türkiye Gümrük Bölgesine Girmesi	45
3.1.5.	Eşyanın Gümrüğe Sunulması	46
3.1.6.	Özet Beyan ve Gümrüğe Sunulan Eşyanın Boşaltılması	46
3.1.7.	Gümrüğe Sunulan Eşyaya Gümrükçe Onaylanmış Bir İşlem veya Kullanım Belirlenmesi Zorunluluğu	47
3.1.8.	Eşyanın Geçici Depolanması:	48
3.2.	Genel Antrepo Yönetim Sistemi Etkileşimli Birimleri	49
3.3.	Genel Antrepo Yönetim Sisteminin Şemaları	50
3.4.	Sistem Analizinden Elde Edilen Sonuç	54
3.5.	Sistem Tasarımı	55
3.6.	Örnek İşlemler	60
	Sonuç ve Genel Değerlendirme	70
	Kaynakça	71
	Özgeçmiş	73

Şekiller Listesi

<u>Şekil No</u>	<u>Sayfa No</u>
Şekil 1: Dosya; kayıtlardan, kayıtlar alanlardan oluşur.	2
Şekil 2: Verinin dört düzeyini ifade eden ANSI/SPARC veri tabanı referans modeli.	7
Şekil 3: Hiyerarşik veri modeli.	10
Şekil 4: Ağ veri modeli.	11
Şekil 5: Varlık-İlişki Diyagramı.	14
Şekil 6: Nesne ilişkisel veri modeli.	18
Şekil 7: Verinin nesne ve tablo görünümü.	20
Şekil 8: Nesne yönelimli veri tabanı ile ilişkisel veri tabanı karşılaştırılması	21
Şekil 9: Özelleştirme/Genelleme ve Kümeleme.	25
Şekil 10: Çoklu kalıtım hiyerarşisi ve birleşik sınıf.	25
Şekil 11: NYVTS’de üç düzey mimarisinin kavramsal görünümü.	27
Şekil 12: NYVM Öğrenci – Ders İlişkisi.	30
Şekil 13: Kullanıcı ve kullanım biçimleri	32
Şekil 14: Birleştirme ilişkisi.	32
Şekil 15: Genelleme ilişkisi.	33
Şekil 16: Mesajın nesneler arasındaki davranışı.	34
Şekil 17: Mesajın nesneler arasındaki davranışı (şartlı).	34
Şekil 18: Malzeme hazırlığı için mesajın hareketi.	35
Şekil 19: Stokta varsa ya da yoksa mesajın hareketi.	35
Şekil 20: Nesneler arası mesajın geçişi (İşbirliği diyagramı).	36
Şekil 21: Şarta göre malzeme hazırlığı (İşbirliği diyagramı).	36
Şekil 22: Durum diyagramında nesnenin tanımı.	36
Şekil 23: Durum diyagramında ilk durum ve şartlar.	37
Şekil 24: Kontrol durumu.	38
Şekil 25: Süper durum diyagramı.	38
Şekil 26: Eylem diyagramı.	39
Şekil 27: Sipariş işlemi için eylem diyagramı.	40
Şekil 28: Yayılma diyagramı.	41
Şekil 29: Bütün olarak fiziksel diyagram.	41

Şekil 30: Veri iletişim yolu.	50
Şekil 31: Fonksiyonlar arası süreç haritası.	51
Şekil 32: Mal kabulü eylem diyagramı.	52
Şekil 33: İthalat organizasyon eylem diyagramı.	53
Şekil 34: İthalat eylem diyagramı.	53
Şekil 35: Sevkiyat eylem diyagramı.	54
Şekil 36: Firma sınıf diyagramı.	56
Şekil 37: Personel sınıf diyagramı.	57
Şekil 38: İrsaliye sınıf diyagramı.	58
Şekil 39: Stok hareket sınıf diyagramı.	59
Şekil 40: Sunucu çalışması.	60
Şekil 41: İstemci giriş.	61
Şekil 42: Referans sınıfları.	62
Şekil 43: Stok kartları kaydetme, okuma, silme.	62
Şekil 44: Giriş irsaliyesi kullanıcı arayüzü.	65
Şekil 45: Object manager nesne görünümü.	66
Şekil 46: Stok giriş.	67
Şekil 47: Sorgulama.	68

Tablolar Listesi

<u>Tablo No</u>	<u>Sayfa No</u>
Tablo 1: ÖĞRENCİ ilişki şeması.	15
Tablo 2: İlişkisel tablolar.	16
Tablo 3: Veri tabanlarının gelişimi.	26

Kısaltmalar Listesi

NYVT	Nesneye yönelik veri tabanı
NYVTYS	Nesneye yönelik veri tabanı yönetim sistemi
a.g.k.	Adı Geçen Kitap
a.g.m.	Adı Geçen Makale
ANSI	Amerikan Ulusal Standartlar Enstitüsü
VTYS	Veri Tabanı Yönetim Sistemi
VTD	Veri Tanımlama Dili (Data Definition Language)
VDD	Veri Düzenleme Dili (Data Manipulation Language)
DKD	Veri Kontrol Dili (Data Control Language)
SD	Sorgulama Dili (Data Query Language)
SPARC	Standartların Planlanması ve Gerekleri Komitesi (Standarts Planing and Requiments Committee)
CODASYL	Veri Tabanı İş Grubu (Conference On Data Systems Languages)
CAD	Bilgisayar Destekli Tasarım
CAM	Bilgisayar Destekli Üretim
BT	Geleneksel Bilişim Teknolojileri
UML	Birleşik Modelleme Dili (Unified Modeling Language)
SQL	Yapısal Sorgu Dili

GİRİŞ

The Standish Group'un yayınladığı The Chaos Report adlı araştırma, ABD'de farklı sektörlerde farklı ölçeklerden firmalarda gerçekleştirilen 30,000 yazılım projesinin ne derece başarılı olduğunu araştırmıştır. The Chaos Report'un 2004 yılı rakamlarına göre bu yazılım projelerinin sadece %34'ü başarılı olmuş; %15'i başladıktan sonra iptal edilmiş; %51'i ise tartışmalı, yani zamanında bitirilememiş veya gereksinimleri karşılamamış projelerdir.¹

Yazılım projelerinin başarısız olma sebepleri araştırıldığında nedenin, öngörülen zamanın aşılması ve gereksinimlerin yeterince belirlenmemesi olduğu görülmektedir. Nesne yönelimli sistemlerin gerçek dünyadaki nesneleri varlık şeklinde modellediği ve proje analizinde gereksinimlerin belirlenmesi aşamasındaki eksiklikleri gidereceği düşünülmüştür.

Çalışmanın birinci bölümünde veri, veri tabanı sistemleri, veri modelleri hakkında kuramsal bilgiler verilmiştir. İkinci bölümde nesne yönelimli yaklaşım ve nesneye yönelik veri tabanı konusunda bilgi verilerek nesne yönelimli modelleme teknikleri anlatılmıştır. Üçüncü bölümde lojistik sektörünün önemli bir halkası olan Genel Antrepolar (Gümrüklü Depo) için yönetim sistem uygulaması yapılarak, genel antrepoların sorunlarından biri olan bilgi paylaşımı sorunu çözülmeye çalışılmıştır. Bu uygulamada nesne yönelimli modelleme yapılarak nesneye yönelik veri tabanı kullanılmaya çalışılmıştır.

¹ <http://www.bilgipark.com/ver.2.0/>

I. BÖLÜM

1. VERİ TABANI SİSTEMLERİNDE TEMEL KAVRAMLAR

1.1. Veri ve Bilgi

Veri, analize temel teşkil edebilecek; herhangi bir anın, faaliyetin, durumun, nesnenin ya da bunların bütününe sayı, harf ve imgeler grubunun biri veya daha fazlası kullanılarak, önceden kabul görmüş kurallar çerçevesinde yapılan tanımının herhangi bir ortamda kayıtlı bulunması halidir.²

Veri, mantıklı düşünmek için temel oluşturan gerçekler olarak tanımlanmaktadır.³

Webster sözlüğü veriyi bilinen ve hayal edilen şeyler olarak tanımlamaktadır. Amerikan Ulusal Standartlar Enstitüsü (ANSI) veriye iki farklı yaklaşım sergilemektedir.

1. İletişime ve insanlar veya otomatik anlamda işlenebilmeye uygun yapısal anlamda kavram ve gerçeklerin bir sunumudur.
2. Manaların karakterler veya analog değerlerde olduğu gibi herhangi bir sunumudur.

Veri, bir ya da daha çok bilgi parçasından meydana gelen bir topluluktur. Bu bilgi parçası, sonuç çıkartma, tartışma, hesaplama, işlem ya da başka bir faaliyetin kaynağı olabilir ve doğru yerde kullanıldığında *bilginin* temelini oluşturur.⁴

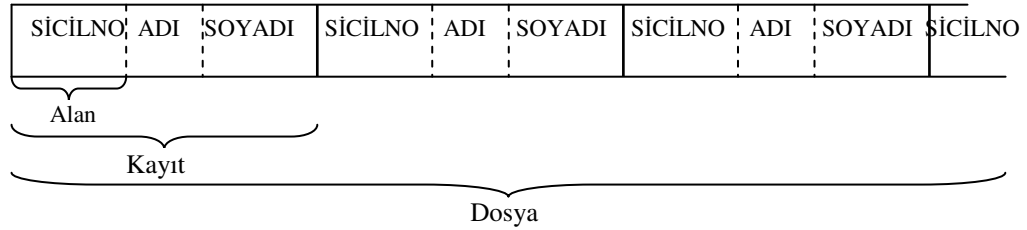
² Babür, Z., *Bilgisayarla İletişim*, Sistem Yayıncılık A.Ş., İstanbul, 1995, s.1

³ Shepherd, J. C., *Database Management*, Irwin, Homewood IL, 1990, s.34

⁴ Stephenes, R., Plew, R., *Sams Teach Yourself Beginning Databases in 24 Hours*, Sams Pub., 2003, s.6

1.2. Dosya, Kayıt ve Alan

Veri saklama birimlerinde depolanan veri topluluklarına *dosya* adını veriyoruz. Dosyalar ise kendi içinde *kayıtlara* bölünmüştür. Kayıtlar da farklı bilgilerden oluşur. Bu bilgilerin her birine *alan* adını veriyoruz.⁵



Şekil 1: Dosya; kayıtlardan, kayıtlar alanlardan oluşur.

Alan adı verilen ayrılmış boşluk, içinde bir veri elemanını saklar. Bu veri elemanı verinin niteliklerini tanımlar. Dosyalar, işlevlerine göre, ana (master) dosya, işlem dosyası, rapor dosyası, iş (geçici) dosyası olarak gruplandırılabilir.⁶

Dosyalar organize edilişlerine ve kayıta erişimlerine göre üç gruptur:⁷

a) Sıralı Dosyalar

Sıralı dosyalarda kayıtlar, bir anahtar veya koda göre sıralama yapılarak dosya içinde sıralı biçimde yer alır. Herhangi bir kayıta erişmek için önceki kayıt veya kayıtlar geçilmiş olmalıdır. Her sıralama için yeni bir dosyanın yaratılması gerekir. Depolama yeri ve işlem zamanının harcanmasına neden olur. En önemli dezavantajı, dosyadaki belirli kayıta doğrudan erişilememesidir.

b) Doğrudan Erişimli Dosyalar

Doğrudan erişimli dosyalarda anahtar alan ile depolama yeri arasındaki ilişkinin doğrulanması temeldir. Böylece istenilen kayıta direkt olarak ulaşılır.

⁵ Özkan, Y., Veri Tabanı Sistemleri, Alfa Ltd.Şti., İstanbul, 2003, s.2-3

⁶ Jefery, D. R., Lawrence, M. J., Systems Analysis and Design, Prentice-Hall of Australia Pty Ltd., 1984, s.6

⁷ Bruch, J., Grundnitski., Information Systems Theory and Practice, John Wiley & Sons INC., Canada, 1989, s.405-409

c) İndeks Sıralı Dosyalar

İndeks sıralı dosyalarda, anahtar alan ile sıralanmış verileri içeren dosya ve veriye ulaşmak için kullanılan indeks dizini içeren dosya olmak üzere iki dosya kullanılır. Bu tür dosyalar, dosya içindeki kayıtlara hem sıralı olarak hem de doğrudan erişmeyi sağlar. İstenilen kayıda erişim, diğer dosyalardaki kadar hızlı değildir. Ancak daha esnektir.

1.3. Veri Yönetimi

Veri yönetimi, veriyi depolama ve depolanan veriye erişme sürecidir. Veri yönetiminin üç temel işlevi vardır:⁸

- Verinin iç ilişkileri ve dış dünya ilişkilerinin belirlenip, standart veri tanımlarının yapılması.
- Belirli bir biçimde verinin depolama araçlarında depolanması.
- Sistem kullanıcıları tarafından depolanmış verinin doğru bilgi sağlayacak şekilde elde edilmesi.

Bilgisayar tabanlı veri yönetiminde iki yaklaşım söz konusudur:

a) Geleneksel dosya işleme ile veri yönetimi

Geleneksel dosya işleme sisteminde, belirli dosyalar, belirli uygulamalarla yaratılır ve yönetilir.⁹ Veri bir programdan diğerine atılır. Her uygulama programı veya sistemi, sadece belirlenmiş bir bölüm veya kullanıcı grubunun ihtiyaçlarını karşılamak için tasarlanan verileri kullanır.¹⁰

⁸ A.g.k., s.397

⁹ Rolland, F. D., *The Essence Of Databases*, Prentice-Hall Europe, Hertfordshire, 1998, s.1

¹⁰ Burch, J., Grudnitski, G., a.g.m., s.400

b) Veri tabanı sistemi ile veri yönetimi

Veri tabanı sistemi ile dosyalar belirli uygulamalarla yönlendirilmez. Veriler sınırsız uygulamalarla paylaşılacak biçimde düzenlenmiştir. Dosyalar arasında ilişkilerin yaratılması söz konusudur.¹¹

Veri tabanı sistemi ile veri yönetiminin, geleneksel dosya işleme ile veri yönetimine göre avantajları:¹²

- a) Veri entegrasyonunu düzenlemek,
- b) Veri erişilebilirliğini arttırmak,
- c) Veri kontrolünü sağlamak ve düzenlemek,
- d) Uygulama geliştirme ve yönetimini kolaylaştırmak,
- e) Veri güvenliğini artırmak.

Bunların dışında aşağıda sıralanan avantajlardan da söz edilebilir.¹³

- a) Veri tabanı sisteminde, aynı veri kaynağının fiziksel olarak, farklı kullanıcılar tarafından paylaşımlı bir şekilde kullanılması,
- b) Veri tekrarının yani aynı verinin birden fazla dosya içinde depolanmasının engellenmesi,
- c) Veri tabanındaki verilerin doğruluğunu ve tutarlı bir bütün oluşturmasını sağlayan kısıtlamaları ifade eden veri bütünlüğünün sağlanması,¹⁴

Veri tabanı sistemi ile veri yönetiminin, geleneksel dosya işleme ile veri yönetimine göre üstünlüğünü gösteren en önemli farklılıklardır.

1.4. Veri Tabanı

Veri tabanını, bir ya da birkaç uygulamada kullanılmak için, gereksiz yinelemelerden arınmış olarak, düzenli biçimde bilgisayar belleklerinde saklanan

¹¹ Rolland, F. D., a.g.m., s.1

¹² Burch, J., Grudnitski, G., a.g.m., s.402-404

¹³ Rolland, F. D., a.g.m., s.2-5

¹⁴ Yarımağan, Ü., Veri Tabanı Sistemleri, Akademi Yay.Ltd. Şti., 2000, s.7

birbirleriyle ilişkili veriler topluluğu olarak tanımlamak mümkündür. Ancak eksik bir tanım olduğu için özelliklerinin sıralanması yerinde olacaktır.¹⁵

- a) Veri tabanı belirli bir kuruluşun birçok uygulamasında kullanılan, birbirleriyle ilişkili, işletimsel verilerden oluşur. Kuruluşun pek çok uygulamasında ortak olarak kullanılan, sürekli verilerdir.
- b) Veri tabanındaki veriler, birden çok uygulama tarafından paylaşılan ortak verilerdir.
- c) Veri tabanında saklanan veriler değişmeyen (statik) veriler değildir. Ekleme, silme ve güncelleştirme işlemleri ile veriler değiştirilebildiği gibi, sorgulama ve raporlama işlemleri de yapılabilir.
- d) Veri tabanındaki veriler üzerinde merkezi bir denetim vardır. Veri tabanı kullanımı yalnız Veri Tabanı Yönetim Sistemi (VTYS) olarak adlandırılan yazılımlarla mümkündür.
- e) VTYS aracılığıyla, veri tabanının bilgisayar belleğindeki fiziksel yapısı kullanıcılardan gizlenir. Kullanıcı sadece istediği bilgilere ulaşmak için isteklerini belirtir.

1.5. Veri Tabanı Yönetim Sistemi

Veri Tabanı Yönetim Sistemi (VTYS), veri tabanı yazılımının bir parçası olarak, verinin nasıl depolanacağı, kullanılacağı ve erişilebileceğini mantıksal olarak yönlendiren bir kurallar sistemidir.¹⁶

VTYS, paylaşılabilen bir kaynak olan veriyi yöneten bir üründür. Bu ürün, kullanıcılara birbiriyle ilişkili verileri gruplama ve paylaşma imkanı vermektedir.¹⁷ VTYS' nin fonksiyonları aşağıdaki gibi özetlenebilir:¹⁸

- a) Sistem içindeki tüm verileri tanımlamak ve veri ilişkilerini belirlemek.
- b) Veri tabanına veri eklemek, veri silmek, değiştirmek ve düzenini sağlamak.

¹⁵ A.g.k., s.1-2

¹⁶ Stephens, R., Plew, R., a.g.m., s.8

¹⁷ Howe, D. R., *Data Analysis for Data Base Design*, Edward Arnold Pty Ltd., Australia, 1983, s.16

¹⁸ Burch, J., Grudnitski, G., a.g.m., s.400-401

- c) Veri kaynağını sağlamlık, güvenilirlik, uygunluk, doğruluk yönünden korumak.
- d) Veri tabanının çoklu kullanıcıları arasında veri paylaşımını sağlamak.
- e) Sistem kullanıcıları tarafından anlaşılabilir bir dil ile veri erişimine izin vermek.

VTYS paketleri iki temel bölümden oluşur:¹⁹

- a) Kullanıcı görünümünü, alanları, kayıt tiplerini ve kayıtlar arasındaki ilişkileri belirleyen “Veri Tanımlama Dili-Data Definition Language (VTD)” ve kaydetme, güncelleme, çağırma veya silme komutlarını içeren “Veri Düzenleme Dili-Data Manipulation Language (VDD)”.
- b) Kullanıcılar üzerinde yetki düzenlenmesini sağlayan “Veri Kontrol Dili-Data Control Language (DKD)” bölümü bulunur. Bu bölümler dışında seçimlik olarak “Sorgulama Dili-Data Qery Language (SD)”, yardımcı programlar ve Veri Sözlüğü gibi bölümlerden söz edilebilir.

VTYS’nin gerçekleştirdiği işlemlerin önemli bir kesimi *Veri Tabanı Yöneticisi* olarak adlandırılan, kullanıcı isteklerini gerçekleştiren ve çok kullanıcıli ortamdaki işletimsel sorunları çözümleyen yazılımlar bütünü olan katman tarafından gerçekleştirilmektedir.²⁰

1.6. Veri Modeli

VTYS, belirli bir veri modeline dayanır. Bir veri tabanı yapısının temelini, veri modeli kavramı oluşturmaktadır. Veriyi mantıksal düzeyde düzenlemek için kullanılan kavramlar, yapılar ve işlemler topluluğuna veri modeli adını veriyoruz.²¹

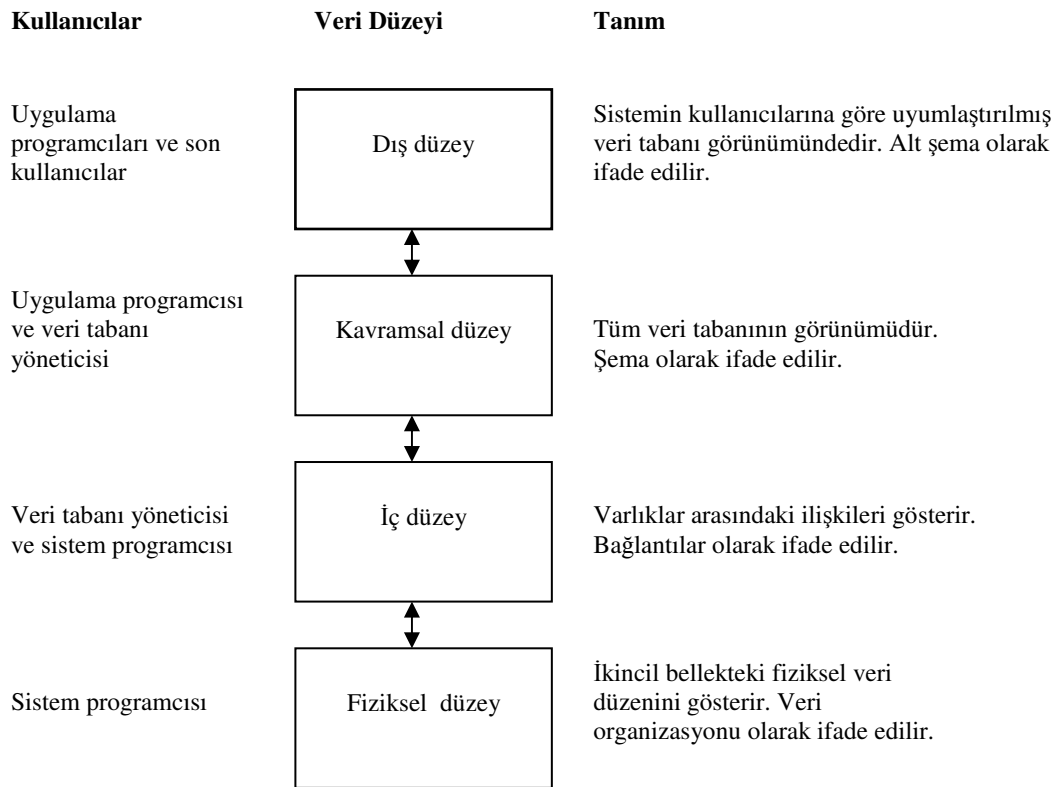
¹⁹ Bontempo, C., Saracco, J., Maro, C., *Database Managemenet Principles and Products*, Prentice Hall Inc., Usa, 1995, s.3

²⁰ Yarımağan, Ü., a.g.m., 5

²¹ Özkan, Y., a.g.m., s.6

1.7. Veri Tabanı Düzeyleri

Veri tabanı düzeylerinin parçalar halinde ortaya konması, VTYS'nin anlaşılabilirliğini sağlar. Amerikan Ulusal Standartlar Enstitüsü (ANSI)'nın Standartların Planlanması ve Gerekleri Komitesi (Standarts Planing and Requiments Committee) tarafından 1975 yılında geliştirilen ANSI/SPARC Veri Tabanı Referans Modeli aşağıda görülmektedir.²²



Şekil 2: Verinin dört düzeyini ifade eden ANSI/SPARC veri tabanı referans modeli.

1.7.1. Dış Düzey

Dış düzey, veri tabanının kullanıcılar tarafından görülen kısmıdır. Alt şema olarak ifade edilen bu görünüm tanımlamaları, veri sözlüğü içinde saklanır ve birçok

²² Burch, J., Grudnitski, G., a.g.m., s.422

uygulama için aynı görünümün kullanılması mümkündür. Görünüm tanımlamaları veri tabanının dış katını oluşturur. Bu kat, veri tabanı ile kullanıcılar arasındaki arayüzdür.²³

1.7.2. Kavramsal Düzey

Veri tabanı sisteminin ana aşaması olan kavramsal düzey, sistem içindeki tüm verilerin tanımını kapsar. Verilerin nasıl depolandığından bağımsızdır. Tüm değişiklikler mantıksal düzeyde meydana gelir ve mantıksal veri bağımsızlığı olarak adlandırılır.²⁴ Veri tabanındaki tüm varlıkları ve varlıklar arasındaki genel ilişkileri kavramsal olarak gösterir. Kavramsal şema oluşturmanın amacı, ihtiyaç duyulan veriyi tam ve doğru olarak göstermek, veri tabanı yapısını oluşturacak kişilerin anlamalarını kolaylaştırmaktır.²⁵

1.7.3. İç Düzey

Bu katta, mevcut kayıtlardaki varlıklar arasında belirli iç ilişkiler (işaretler, indeksler kullanılarak) tanımlanır.²⁶

1.7.4. Fiziksel Düzey

En alt düzeyi oluşturan fiziksel düzey, veri dosyalarındaki kayıtların gruplanmasını ve bu dosyaların veri organizasyonunu içerir. Aslında fiziksel veri tabanı olarak adlandırılan bu katman da kendi içinde dosyalar ve mantıksal kayıtlar düzeyinden başlayıp, fiziksel adresler düzeyine kadar uzanan birçok soyutlama düzeyinde görülebilir.²⁷

²³ Rolland, F. D., a.g.m., s.4-5

²⁴ A.g.k., s.7

²⁵ Burch, J., Grudnitski, G., a.g.m., s.430-431

²⁶ A.g.k., s.423

²⁷ Yarımağan, Ü., a.g.m., 12

1.8. Veri Modelleri

Veriyi mantıksal düzeyde düzenlemek için kullanılan kavramlar, yapılar ve işlemler bütününe *veri modeli* denir.²⁸ Bir veri tabanı yönetim sistemi, bir veri modeli üzerine inşa edilir.²⁹ Veri modelleri, veri tabanındaki ilişkileri soyut bir şekilde canlandırdığından kavramsal düzeyde oluşturulur.³⁰

Veri tabanının oluşturulması amacıyla geçmişten günümüze kadar bir takım farklılıklarla çeşitli veri modelleri geliştirilmiştir. Her biri, bir öncekinin eksikliğini tamamlamak amacına yöneliktir.

1.8.1. Basit Veri Modelleri

Geliştirilen ilk iki model, diğerlerine göre basit bir yapı oluşturduğu için bu modelleri basit veri modelleri olarak adlandırmamız yerinde olacaktır.

1.8.1.1. Hiyerarşik Veri Modeli

1968’de IBM’in müşterilerini bilişim yönetim sistemine dahil ederek, dosya yönetiminde veri tabanı yaklaşımının yararlarını ortaya koymak amacıyla oluşturduğu modeldir. İlişkiler, hiyerarşik olarak sadece tek yönlü meydana gelir.³¹

Hiyerarşik veri modelinde, kayıtlar, ağaç mimarisi şeklinde sıralanıyordu. “Kök” adı verilen kaydın bir ya da daha fazla çocuk kayıtları vardı.³² Kök, modelin en üst düğümüdür. Bir düğüm, altındaki birden fazla düğüme bağlanabilir, fakat, sadece üstündeki tek bir düğüme bağlanabilir. Yani birden-birçoğa ilişkisi vardır.³³

²⁸ A.g.k., s.17

²⁹ Senn, J. A., Analysis & Design of Information Systems, MacGraw-Hill Book Co., Singapore, 1989, s.605

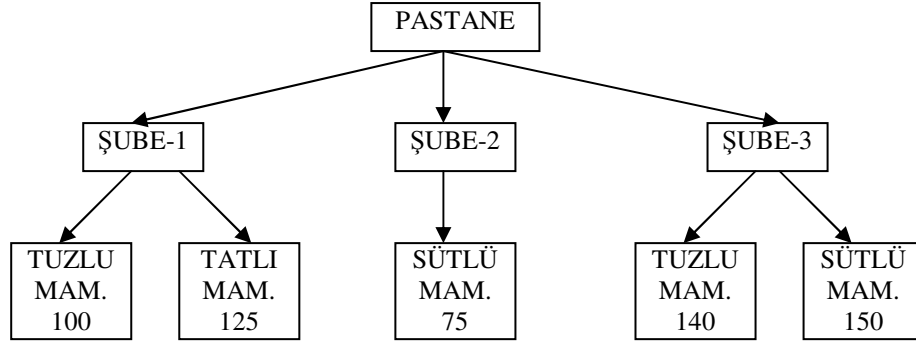
³⁰ Burch, J., Grudnitski, G., a.g.m., s.431

³¹ Rolland, F. D., a.g.m., s.9

³² Stephens, R., PLew, R., a.g.m., s.58

³³ Burch, J., Grunditski, G., a.g.m., s.431

Hiyerarşik veri tabanında, bir düğümün silinmesi gerektiğinde ona bağlı olan alt düzey düğümler de silinmek zorundaydı. Veri tekrarı nedeniyle yer sorunuyla karşılaşılmaktaydı ve esneklik açısından uygun değildi.³⁴



Şekil 3: Hiyerarşik veri modeli.

1.8.1.2.Ağ Veri Modeli

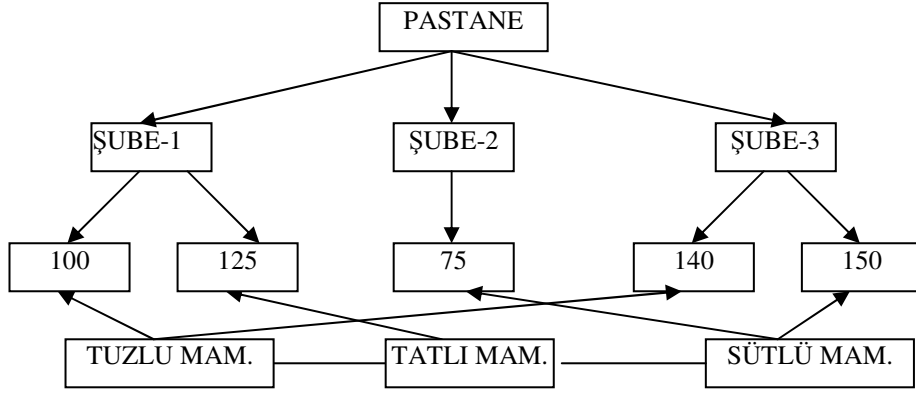
1970’lerde Veri Tabanı İş Grubu (CODASYL) kurularak hiyerarşik modelin geliştirilmesine yönelik çalışmalar başlatılmış, 1975’te bu yeni model CODASYL ya da ağ modeli olarak rapor şeklinde sunulmuştur.³⁵

Ağ veri modeli, hiyerarşik modele benzer. Ağ veri tabanları birden çok ağaç içerebilir. Çoktan-çoğa ilişkileri destekler. Hiyerarşik modelden ayıran en önemli özellik, kök kayıt tipi ihtiyacının ortadan kalkması ve kayıtlar arasında oluşturulan bağlantıların yönü ve sayısında sınırlama olmamasıdır. Ancak, istenen bilgiyi okumak için kayıt mimarisinin bilinmesi ağ modelinde de gereklidir.³⁶

³⁴ Rolland, F. D., a.g.m., s.9

³⁵ Rolland, F. D., a.g.m., s.10

³⁶ Stephens, R., Plew, R., a.g.m., s.60-61



Şekil 4: Ağ veri modeli.

1.8.2. Modern Veri Modelleri

Zamanla bilgisayarda bir işlem yapabilmek için sadece verinin işlenmesi yeterli olmamaya başladı. Ağ veri modelinde, ara varlıklar kullanılarak çoklu ilişkileri göstermek mümkündür. Ancak, bu durum karmaşık bir modelin ortaya çıkmasına ve entegrasyonda sorunlara neden olmaya başladı. Bu nedenle veri ilişkilerinin daha kolay, anlaşılır ve esnek olabilmesi amacıyla yapılan çalışmalar modern veri modellerinin ortaya çıkmasını sağladı.

1.8.2.1. Varlık-İlişki Veri Modeli

Varlık-ilişki veri modeli, 1976'da P.Chen tarafından semantik bir model olarak tanımlanmış ve daha sonra geliştirilmiştir. Varlık-ilişki veri modelinin ana bileşenleri şunlardır:³⁷

a) Varlıklar

Var olan ve ayırt edilebilen her şeye, her nesneye *varlık* denir. Aynı türden benzer varlıkların oluşturduğu kümeye *varlık kümesi* denir.³⁸

³⁷ Rolland, F. D., a.g.m., s.22-24

³⁸ Yarımağan, Ü., a.g.m., s.18

b) Nitelikler

Bir varlık kümesindeki varlıkların özelliklerini göstermek ve varlıkları birbirinden ayırt etmek için *nitelikler* kullanılır. Başka bir deyişle her varlığa belirli sayıda nitelik değeri eşlenir.³⁹

Her niteliğin aldığı değerlere *etki (değer) alanı* adı verilir. O halde nitelik, matematiksel anlamda bir fonksiyon olarak da değerlendirilebilir. Bir varlık kümesinden bir etki alanına eşlendirmeyi sağlar. Böylece her varlık, nitelik ve veri değeri olarak veri çiftleri biçiminde gösterilebilir.⁴⁰

(nitelik, etki alanı)

(Malzeme No, 315)

Nitelikler, türetilen nitelik, çok değere sahip nitelik ve birleşik nitelik şeklinde olabilir.⁴¹

Bir nitelik kullanılarak başka bir varlık elde edilebiliyorsa, bu yeni niteliğe *türetilen nitelik* adı verilir.

Sayaç son okuma tarihi =====> Suyu kullanma gün sayısı

Bir nitelik birden fazla değer ile eşlenebiliyorsa, *çok değere sahip nitelik* adı verilir.

Kullanılan renkler =====> Yeşil

=====> Sarı

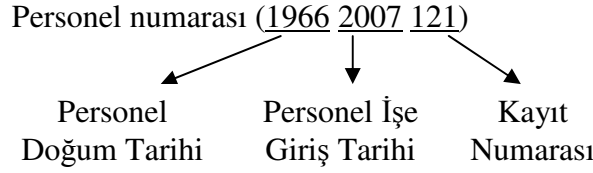
=====> Beyaz

³⁹ A.g.k.

⁴⁰ Özkan, Y., a.g.m., s.11

⁴¹ A.g.k., s.11-12

Birden fazla nitelik birleştirilerek yeni bir nitelik oluşturulabilir. Bu tür niteliklere *birleşik nitelik* adı verilir. Örneğin bir personelin numarası şu şekilde oluşturulabilir.



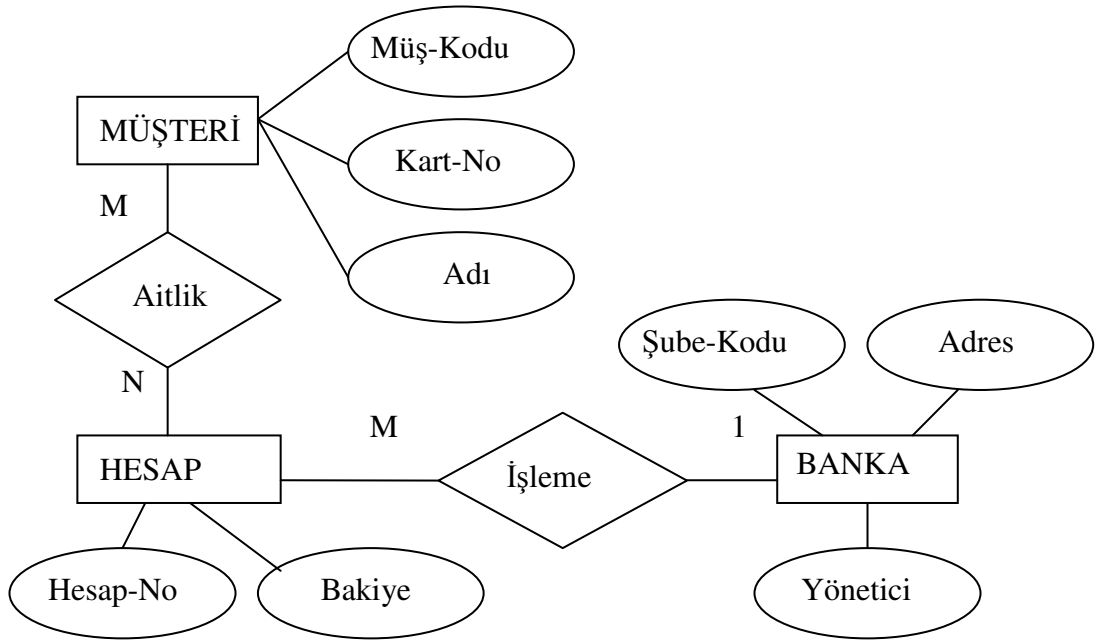
Bir ilişki, varlığın bir fonksiyonu ise buna *rol* denir.

c) İlişkiler

Varlıklar arasındaki etkileşimi gösterir. Bir varlıkla ilişkilendirilebilecek varlıkların sayısına *eşleme sayısı* diyoruz. Eşleme sayıları $n \geq 2$ varlık için söz konusudur ve ikili ilişkilerin konulması açısından yararlıdır. A ve B gibi iki varlık kümesi arasındaki R ilişki kümesi için eşleme durumları şu şekilde ifade edilir:⁴²

- A varlık kümesi içindeki bir varlık, B varlık kümesi içindeki sadece bir varlık ile ve B varlık kümesi içindeki bir varlık A varlık kümesi içinde sadece bir varlık ile ilişkili ise *birden-bire ilişki* söz konusudur.
- A kümesi içindeki bir varlık B kümesi içindeki birden fazla varlık ile ilişkili ise bu ilişkiye *birden-çoğa ilişki* adı verilir.
- A varlık kümesindeki birden fazla varlık B kümesindeki bir varlık ile ilişkili ise bu eşlemeye *çoktan-bire ilişki* denir.
- A varlık kümesi içindeki bir varlık, B varlık kümesi içindeki birden fazla varlık ile ilişkili ise ve B varlık kümesi içindeki bir varlık, A varlık kümesi içindeki birden fazla varlık ile eşleniyorsa *çoktan-çoğa ilişki*den söz edilir.

⁴² A.g.k., 13-19



Şekil 5: Varlık-İlişki Diyagramı.

Varlıklar ve varlıklar arasındaki ilişkilerin gösteriminde *varlık-ilişki diyagramları* kullanılır.⁴³ Varlıklar isimleri, nitelikler sıfatları, ilişkiler de fiilleri gösterir. Dikdörtgenler varlık kümeleri, halkalar nitelikler, baklava şekilleri ilişkilerdir.⁴⁴

Varlık veya ilişki kümesi içinde yer alan varlıkların birbirinden ayırt edilebilmesi için iki türlü anahtar kullanılır. Varlık kümesi içinde yer alan varlığı kesin olarak tanımlamaya yarayan anahtara *süper anahtar*, bir süper anahtarın herhangi bir alt kümesi aynı zamanda bir süper anahtar değil ise bu anahtara da *aday anahtar* ya da *anahtar* denir. Bazı durumlarda varlık kümesi içinde bir anahtar bulmak ya da oluşturmak mümkün olmaz. Bu varlık kümelerine *zayıf varlık kümesi* denir. En az bir anahtar oluşturulabiliyorsa bu kümelere *kuvvetli varlık kümesi* denir.⁴⁵

⁴³ Hawryszkiewicz, I., Introduction to Systems Analysis and Design, Prentice-Hall of Australia Pty. Ltd., 1998, s.184

⁴⁴ Burch, J., Grudnitski, G., a.g.m., s.423-424

⁴⁵ Özkan, Y., a.g.m., s.20-22

1.8.2.2. İlişkisel Veri Modeli

İlişkisel veri tabanları ile ilgili özellikler 1970’lerde E.F. Codd tarafından ortaya konmuştur. İlişkisel yaklaşımda veri, kavramsal aşamada satır ve sütunları içeren iki boyutlu tablolar içinde gösterilir. Tablolar ilişkileri ifade eder. Dosya içindeki kayıtları ifade eden satırlardan ve her satırdaki nitelik değerini gösteren sütunlardan oluşur.⁴⁶

Tablonun diğer bir deyişle ilişkinin özellikleri:⁴⁷

- a) Satırların sırası önemsizdir.
- b) Tüm satırların birbirinden farklı olması gerekir.
- c) Tüm satırlar birbirinden farklı olduğuna göre ilişkinin en az bir anahtarı vardır.
- d) İlişkinin her sütunu bir niteliğe karşılık geldiğine ve sütun başlığında niteliğin adı yer aldığına göre, sütunların sırası da önemsizdir.
- e) Bir sütunda yer alan tüm değerler, belirli bir değer alanından gelen değerler olup tümünün türü aynıdır.
- f) Tablonun nitelikleri yalın ve tek değerli niteliklerdir.

İlişkinin adı ve ilişki niteliklerinin adlarından oluşan tanım deyimine *ilişki şeması* denir.

Tablo 1: ÖĞRENCİ ilişki şeması.

ÖĞRENCİ(Öğr-No, Öğr-Adı,Öğr-Adres)
ÇALIŞMA(Öğr-No.,Ders-Kodu)
DERS(Ders-Kodu,Ders-Adı)

⁴⁶ Rolland, F. D., a.g.m., s.10-11

⁴⁷ Yarımağan, Ü., a.g.m., s.41

Tablo 2: İlişkisel tablolar.

ÖĞRENCİ

Öğr-No	Öğr-Adı	Öğr-Adres
117 142 235	Zafer Tan Ayla Acar Şevket Bahar	Güzel Sk. No1. Defne Sk. No.47 Seher Sk. No.23

ÇALIŞMA

Öğr-No	Ders-Kodu
117 142 235	B3 M7 B3

DERS

Ders-Kodu	Ders-Adı
B3 M7 T6	Biyoloji Matematik Türkçe

İlişkisel veri tabanının tasarlanması aşamasında, verinin tekrarlanması, kaybını veya yetersizliğini önlemek için *normalleştirme* işlemi uygulanır. Normalleştirme teorisi normal form kavramına dayanır.⁴⁸

İlişkisel veri modeli, gerçek dünya ile ilgili verileri ifade edebilmek için birbirleriyle bağımlı kümeler (fonksiyonel bağımlılık) kavramını kullanır. Model olgunlaştıkça fonksiyonel bağımlılıktan daha ileri bağımlılıklar da ortaya çıkabilir.⁴⁹

Codd'a göre fonksiyonel bağımlılık göz önüne alınmadan oluşturulan bir ilişkide güncelleştirme anormallikleri bulunmaktadır. Bunları ortadan kaldırmak için

⁴⁸ Özkan, Y., a.g.m., s.43

⁴⁹ Hurson, A.R., *Paralel Architectures for Database Systems*, IEEE Computer Society Press Washington D.C., 1989, s.39

birinci dereceden normalleştirme, yani “normal form” yapılmış, daha sonra gerek görüldükçe normalleştirme daha üst derecelere çıkarılmıştır.

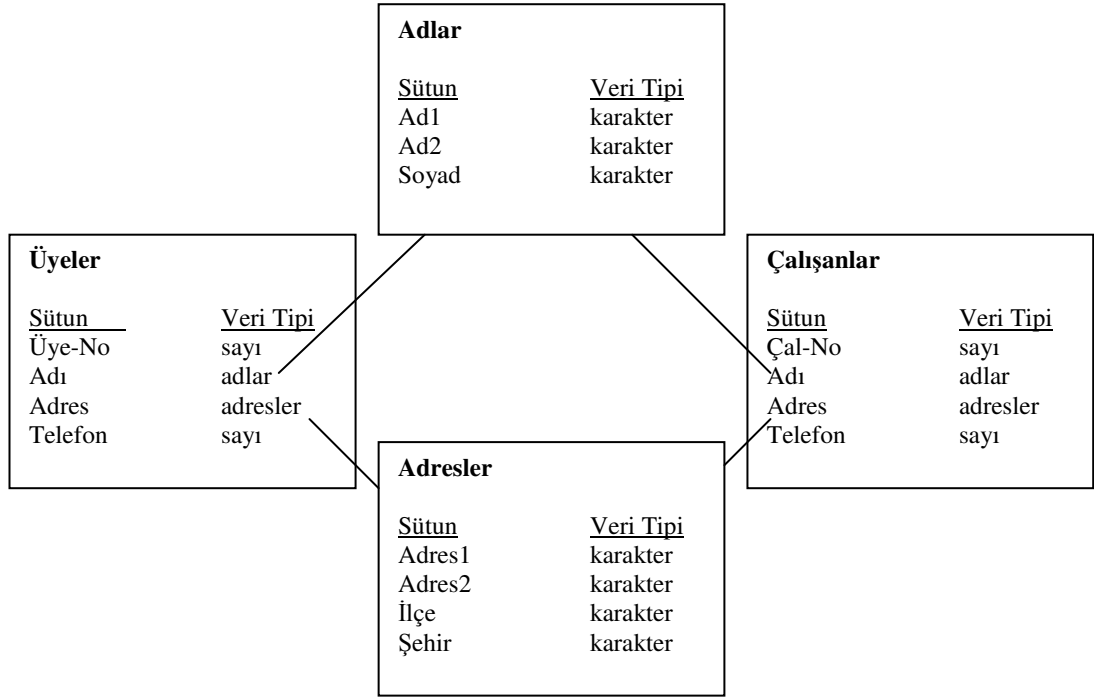
1.8.2.3.Nesne İlişkisel Veri Modeli⁵⁰

Nesne ilişkisel veri tabanı, ilişkisel işlevselliğin üzerinde nesne yönelimli özellikler içerir. Bu türdeki ilk veri tabanı 1997’de oluşturulan Oracle8’dir. Desteklediği veri tipleri sayılar, tarihler ve karakterlerin dışında kullanıcının oluşturabileceği veri tiplerine kadar genişletilmiştir.

Nesne ilişkisel veri tabanı sisteminin amacı, ilişkisel tipteki bir ortamda nesne tabanlı veri depolamayı sağlamaktır. Bu sistem, ilişkisel veri tabanı yönetim sisteminin nesne desteğini içeren melez bir veri modeli olarak kabul edilir.

İlişkisel veri modelinde tamsayı, karakter katarı, reel sayılar gibi basit veriler tablolarda saklanırken, nesne ilişkisel veri modelinde veri depolanan tabloların yerini nesneler alır. Yapılan sorguların sonuçları satırlardır. Bu işlevselliği nesnelerle sağlamak ve bağlantısız verilerle nesne yönelimli programlama dillerini bütünleştirmek için ilişkisel modelde değişiklik yapılması gerekir. Bir VTYS’nin nesne ilişkisel yeteneklere sahip olabilmesi için, kullanıcı tanımlı veri tipleri, nesne ve kalıtım desteği verebilmelidir.

⁵⁰ Stephens, R., Plew, R., a.g.m., s.67-68



Şekil 6: Nesne ilişkisel veri modeli.

1.8.2.4.Nesne Yönelimli Veri Modeli

Veri tabanı yönetim sistemleri uygulamasında klasik veri modelleri için yapılan en önemli eleştiri, veri kümelerinin düşünce olarak bütünleştirilebilmesi, ancak uygulamada yetersizliklerin ortaya çıkmasıydı. Çünkü, gerçekten verinin nasıl davrandığı modellemede yer almamaktaydı. İşte nesne yönelimli veri modeli, semantik ifadelerin bütünleşmesiyle bu eksiklikleri gidermeye çalışır.⁵¹

Kullanılmakta olan ilişkisel veri tabanı teknolojisi, endüstride pek çok uygulama için oldukça yeterlidir. Ancak, ilişkisel veri tabanı teknolojisi, nesne yönelimli diller tarafından sağlanan veri depolama modellerini hesaba katmaz. Özellikle, ilişkisel veri tabanlarının nesne yönelimli uygulamalarla birlikte kullanılması sırasında ortaya çıkan sorunlar vardır. Nesneye yönelik veri tabanı teknolojisi bu sorunları ortadan kaldırmak için geliştirilmiştir.⁵²

⁵¹ Rolland, F. D., a.g.m., s.16-23

⁵² Stephens, R., Plew, R., a.g.m., s.134

Üçüncü kuşak veri tabanı sistemleri olarak adlandırılan bu sistemler 1990'lı yılların sonuna doğru ticari olarak pazarlanmaya başlanmıştır.⁵³

⁵³ Yarımağan, Ü., a.g.m., s.201

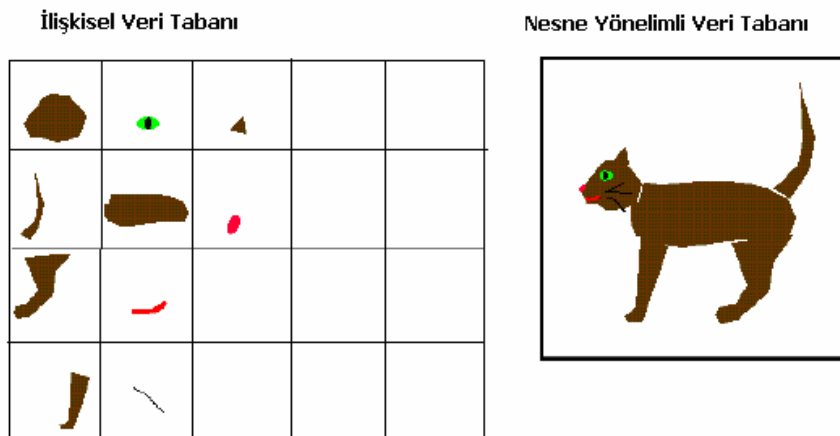
II. BÖLÜM

2. NESNEYE YÖNELİK VERİ TABANI

Nesne yönelimli sistemler, farklı sistemler ve metodolojiler için kullanılmıştır. Genel olarak bu sistemler gerçek dünyadaki nesne denilen şeyleri varlık şeklinde modellemeyi temel almaktadır. Nesneler, ortak karakteristikler içeren sınıflar içerisinde bulundurulur. Sınıf hiyerarşisi kavramı ise insanların nesneleri küresel anlamda kategorize etmelerini sağlamak için geliştirilmiştir.⁵⁴

2.1. İlişkisel Veri Tabanı ve Nesneye Yönelik Veri Tabanı

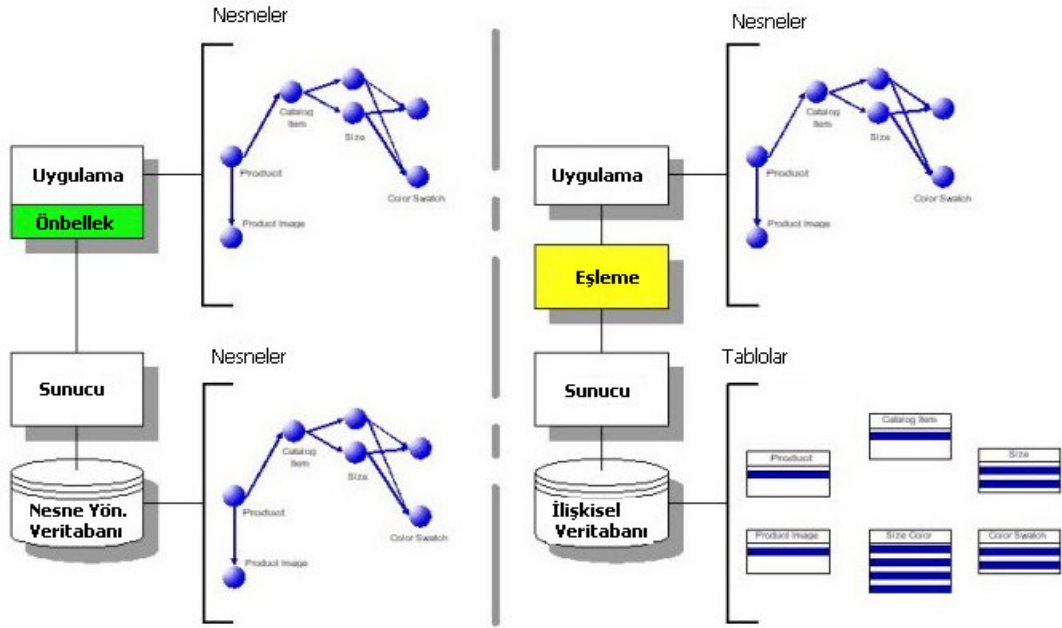
Nesneye yönelik veri tabanı yönetim sistemi (NYVTYS), “kapsülleme” gibi uygun giriş metotlarıyla birlikte veriyi saklar. İlişkisel veri tabanında ise veri normalleştirme ile uygun şekilde saklanır. Basit olarak anlatmak gerekirse, ilişkisel veri tabanında bir nesnenin her parçası tabloda ayrı bir hücrede tutulur ve tablo o nesneyi oluşturur. Oysa NYVTYS’nde nesne bütün olarak tek bir hücrededir. Burada bilgisayar destekli tasarım (CAD) uygulamalarında olduğu gibi karmaşık veri tipleri depolanabilir. Multimedya uygulamalarında olduğu gibi aynı veri tabanı içindeki veri tipleri genişletilerek dizilebilir. Evrimsel uygulamalarda olduğu gibi zamansal olarak nesnelerin takibi daha kolaylaşır.⁵⁵



Şekil 7: Verinin nesne ve tablo görünümü

⁵⁴ Bukhres, O.A., Elmagarmid, A.K., *Object-Oriented Multidatabase Systems*, Prentice Hall Inc., New Jersey, 1996, s.7

⁵⁵ <http://www.dis.port.ac.uk/~chandler/OOlectures/database/database.htm>, 2002



Şekil 8: Nesne yönelimli veri tabanı ile ilişkisel veri tabanı karşılaştırılması.

NYVT nesneleri farklı veri yapılarından herhangi bir eşleme yapılmadan kaydetmeye izin verir. İlişkisel veri tabanında nesnelerden tablolara eşleme yapmak gerekir. Bu eşleme uyumsuzluğu *özdirenç hatası* olarak adlandırılır. Şeklin sol tarafında nesnelerin doğrudan kaydedildiği gösterilmektedir. NYVT’de özdirenç hatasının olmaması özellikle karışık verilerin kaydedilmesinde önemli derecede performans artırır.⁵⁶

2.2. Nesneye Yönelik Yaklaşım

Nesneye yönelik yaklaşımın dayandığı temel ilkeler şu şekildedir:⁵⁷

- Bir uygulamadaki her varlık bir nesne olarak modellenir. Her nesnenin bir kimliği vardır ve nesne değerlerinden farklıdır. Böylece, bir nesne başka nesnelerden oluşabilir; bir nesneye birden çok nesne gönderme yapılabilir, yani, bir nesne birden çok nesne tarafından paylaşılabilir.

⁵⁶ <http://www.soc.napier.ac.uk/module.php3?op=getresource&cloaking=no&resourceid=7462390,Objekt-Oriented Databases, Module No: CO42009, Group Seminar: OODB System Comparison>

⁵⁷ Yarımağan, Ü., a.g.m., s.203-205

- b) Her nesnenin yapı ve davranış bilgileri vardır ve bu bilgiler kapsüllenir. Bir nesnenin yapısı, olgu (örnek) değişkenleri tarafından belirlenir. Olgu değişken değerleri skalar değer, küme, birleşik değer veya bir başka nesneye referans olabilir. Olgu değişkeni değerleri zamanla değişen değerlerdir. Bir nesnenin olgu değişkenleri nesnenin bir andaki durumunu belirler. Nesnenin davranışı ise nesne üzerinde uygulanabilecek yöntemlerden (metotlardan) oluşur.
- c) Bir nesnenin durumuna bir ileti ile ulaşılabilir. Nesne tarafından bir iletinin alınması ilgili yöntemin işletiminin başlatılmasına yol açar. Kısaca nesneler arası iletişim, ileti alışverişli ya da ileti aktarma ile gerçekleşir.
- d) Benzer yapı ve davranışa sahip nesneler gruplandırılarak sınıflar oluşturulur. Genelde, her nesne sınıfı bir sınıfın örneğidir. Nesneler arası ileti aktarma, sınıflar düzeyinde yapılan tanımlara uygun olarak gerçekleştirilir.
- e) Sınıflar arasında hiyerarşik bir düzenleme vardır. Bu düzenlemede her sınıfın altında bir ya da birkaç alt sınıf bulunabilir. Diğer taraftan her sınıf bir ya da birkaç üst sınıfın altında yer alabilir. Her sınıfın yapı ve davranış özellikleri kalıtım yoluyla alt sınıflara geçer. Eğer bir sınıfın kendi yapı ve davranışı varsa, bunlar kalıtım yoluyla gelen yapı ve davranışlara üstünlük sağlar. Yukarıda belirtilen kalıtım kurallarının istisnaları olabilir. Diğer taraftan çoklu kalıtım durumunda çatışmalar oluşabilir. Sistem tanımında, çatışmaların nasıl çözümleneceğinin belirtilmesi gerekir.

2.3. Nesne Yönelimli Veri Modelinin Özellikleri

a) Nesne Kimliği

Nesne yönelimli sistemde her nesne, diğer nesnelerden ayırtedilebilen bir kimliğe sahiptir.⁵⁸ Nesne kimliğini oluşturmak için ilişkisel modeldeki birincil

⁵⁸ Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., Object-Oriented Modeling And Design, Prentice-Hall, New Jersey, 1991, s.24

anahtar gibi *değer yaklaşımı*, kullanıcı tarafından sağlanan *adlandırma yaklaşımı* ve otomatik olarak *sistem tarafından üretme yaklaşımı* kullanılır.⁵⁹

b) Karmaşık Nesneler

Bileşen olarak başka nesneleri içeren nesnelere *karmaşık nesne* denir. Nesneler arası içirme hiyerarşisi kullanılarak bir nesnenin içerdiği ve bir nesnenin içinde yer aldığı tüm nesneler bulunabilir. Her kullanıcı, içirme hiyerarşisinin belirli bir alt ağacı üzerinde işlem yapabilir ve belirli bir nesneyi, içerdiği tüm nesnelerle birlikte, bir bütün olarak görebilir ve bu nesneye bir bütün olarak gönderme yapabilir.⁶⁰

c) Türler ve Sınıflar

Programlama dilleri alanından alınan tür ve sınıf kavramları, NYVT’da soyut düzeyde değer ve nesne ayrımını yansıtmaktadır. *Tür*, bir değer kümesinin zaman-bağımsız tanımıdır. Değer kümesindeki değerler karmaşık değerler de olabilir. Tür ile sınıfın yalnız yapısal kesimi tanımlanır. *Sınıf* kavramı ile tür ve davranış bilgileri birleştirilir (kapsülendir). Davranış bilgileri arasında özellikle nesnelerin yaratılmasını ve silinmesini gerçekleştiren işlemler yer alır.⁶¹

d) İşlemler ve Metotlar

İşlem, bir sınıf içindeki nesneler tarafından yerine getirilen bir işlev ya da dönüşümdür. Bir nesnenin davranışı, ait olduğu sınıfın metotları tarafından belirlenir. *Metot*, bir sınıfın bir işlemi uygulama biçimidir. Nesnelere uygulanan metotlarla işlemler yerine getirilir. Nesnelerin davranışı, belirlenen metotlara göre gerçekleşir ve işlem tamamlanır. Nesne yönelimli veri modellerinde *kapsülleme* önemli bir kavramdır. Nesne ile ilgili olan her şey sınıf tanımı içinde kapsüllenmiştir. Sınıf tarafından tam olarak tanımlanmış davranış, sadece dışarıdan mesajların gönderilmesini sağlayan arayüz ile gerçekleşir.⁶²

⁵⁹ Yarımağan, Ü., a.g.m., s.205

⁶⁰ A.g.k., s.206

⁶¹ A.g.k.

⁶² Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., s.25

e) *Kalıtım*

Kalıtım, üst ve alt türlerde önemli benzerlikleri gösteren ve nesne yönelimli programlamada çok önemli bir kavramdır. Bir nesne sınıfının, diğer bir nesne sınıfından miras aldığını söyleyebiliriz. Örneğin, “Kişi” sınıfının tüm işlem ve metotları, “Kadın” ve “Erkek” sınıflarının işlem ve metotları olarak miras kalır.⁶³

- *Özelleştirme / genelleme ve kümeleme*

Her veri tabanı şeması, gerçek dünyadaki bir uygulama alanını, belirli bir soyutlama düzeyinde tanımlar. Gerçek dünyadaki uygulama alanlarında da birbirleriyle bağlantılı nesneler yer alır. Bu bağlantıları şema düzeyinde göstermek için *özelleştirme/genelleme ve kümeleme* kavramlarından yararlanılır.⁶⁴

Aynı niteliklere sahip nesne türlerinin genel bir tür içinde birleştirilmesine *genelleme* adı verilir. Bunun tersi ise yani genel bir türün aynı niteliklerle belirlenen farklı türlere ayrılmasına *özelleştirme* denir.⁶⁵ *Kümeleme* ise nesne türünü oluşturan nesne ve türlerin birer nitelik olarak soyutlanmasıdır.⁶⁶

Özelleştirme/genelleme ile **ÇEŞİDİDİR (A-KIND-OF)** ya da **ODUR (IS-A)** ilişkisi, kümeleme ile **PARÇASIDIR (A-PART-OF)** ilişkisi tanımlanır.⁶⁷ Semantik modellemede bir nesne türü değeri, başka bir nesne türünün bir örneği olarak o sınıfın içinde yer almışsa **ÖRNEĞİDİR (INSTANCE-OF)**⁶⁸ veya **TİPİDİR (TYPE-OF)**⁶⁹ ilişkisi ortaya çıkar.

⁶³ Rolland, F. D., a.g.m., s.152

⁶⁴ Yarımağan, Ü., a.g.m., s.207

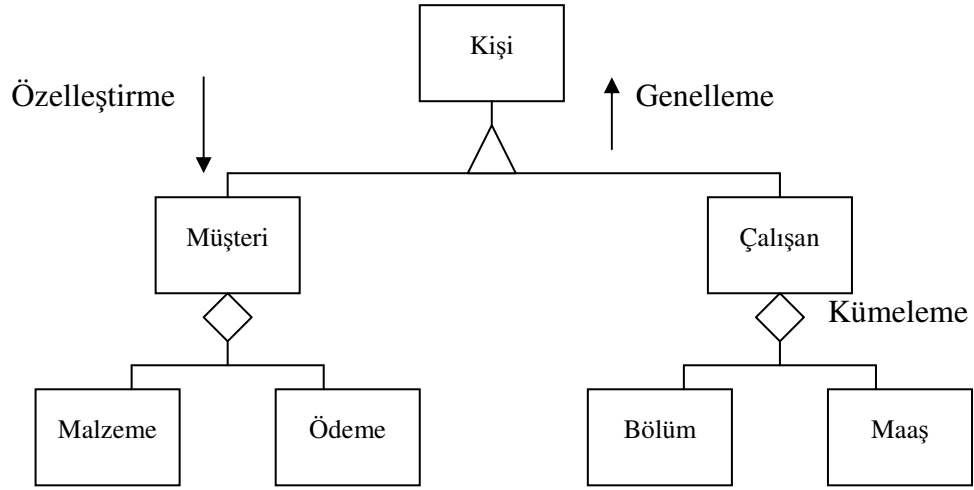
⁶⁵ Rolland, F. D., a.g.m., s.20-21

⁶⁶ Bekke, J. H., *Semantic Data Modeling*, Prentice Hall International Ltd., Hertfordshire, 1992, 88

⁶⁷ Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen W., s.58-59

⁶⁸ Bekke, J. H., a.g.m., s.87

⁶⁹ Rolland, F. D., a.g.m., s.21-22

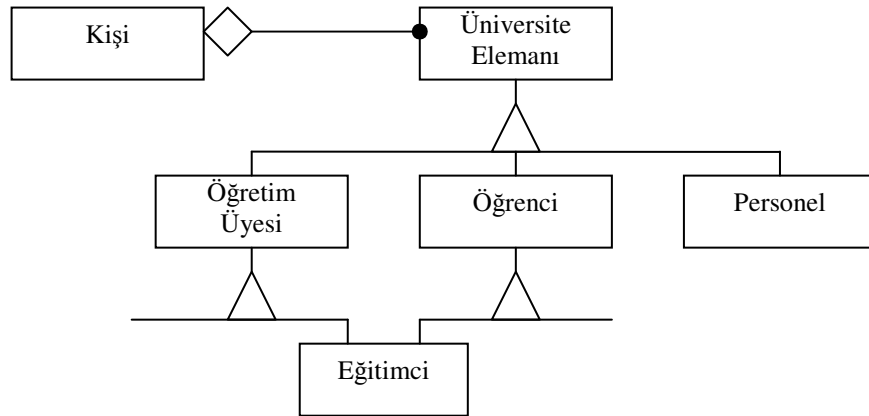


Şekil 9: Özelleştirme/Genelleme ve Kümeleme.

“Çalışan sınıfı” ve “Müşteri sınıfı”, “Kişi sınıfı”nın alt sınıflarıdır. Bu nedenle;

“Çalışan **ODUR (IS-A)** Kişi veya Çalışan **ÇEŞİDİDİR (A-KIND-OF)** Kişi” şeklinde genelleme yapılır. Tersine düşünüldüğünde **Kişi** sınıfı, **Müşteri** ve **Çalışan** sınıfları olarak özelleştirilmiştir. **Çalışan** sınıfı, **Bölüm** ve **Maaş** sınıflarından oluşur. Bu nedenle; “**Bölüm PARÇASIDIR (A-PART-OF)** Çalışan” şeklinde gösterilir. **Bölüm** ve **Maaş** sınıfları kümeleme ile **Çalışan** sınıfını oluşturmuştur.

Bir sınıf, birden fazla üst sınıfın bütün kalıtsal özelliklerini taşıyorsa *çoklu kalıtım* vardır ve birden fazla üst sınıfla oluştuğu için *birleşik sınıf*tır.



Şekil 10: Çoklu kalıtım hiyerarşisi ve birleşik sınıf.

2.3.1. Nesneye Yönelimli Veri Tabanlarının Gelişimi

Nesneye yönelik veri tabanı sisteminin yaygın şekilde ilk kullanıldığı alanlar, CASE, CAD ve CAM olmuştur. Bugün ise telekomünikasyon, sağlık, mali işler, multimedya, metin, dokümantasyon, kalite yönetimi gibi alanlarda kullanılmaktadır. Geleneksel bilişim teknolojileri (BT) metotları ile çalışmayan gelecek nesil uygulamaları vardır. Ancak işyerlerinde verinin %90'ı kağıt-tabanlı sistemlerde saklanır. Çünkü verilerin büyük bölümü kayda uygun değildir. Veri tabanlarının tarihsel gelişimi de her birinin bir öncekine göre üstünlüklerine bakılarak özetlenebilir.⁷⁰

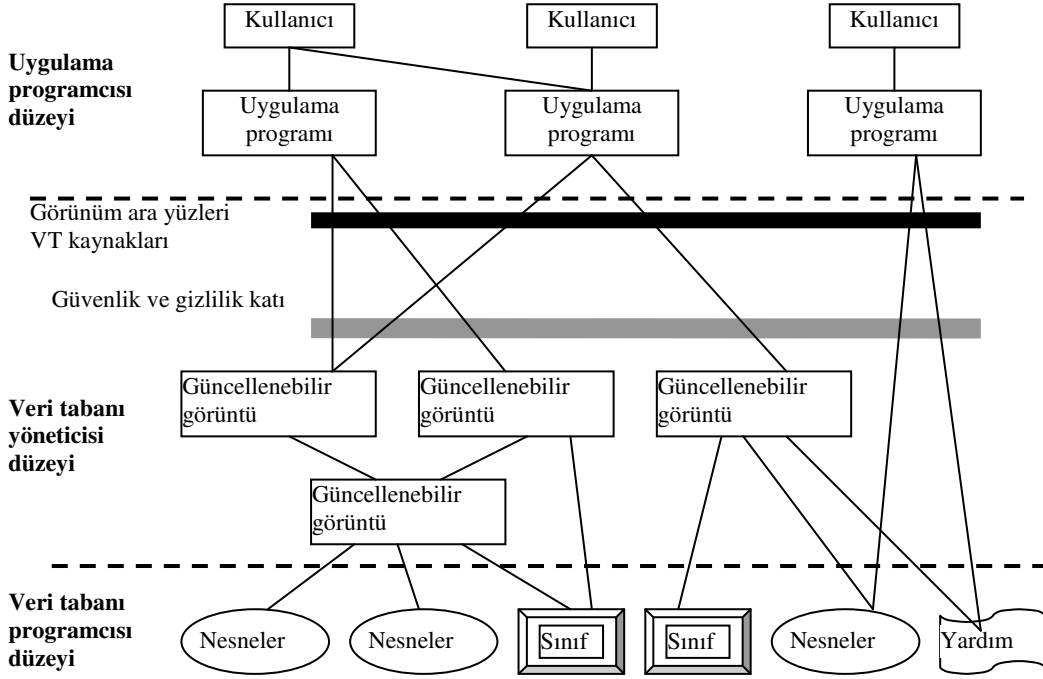
Tablo 3: Veri Tabanlarının Gelişimi

Düz Dosya Sistemi (1950' ler)	* Daha sonra işlem yapmak üzere sürekli veri depolamak
Hiyerarşik / Ağ (1960' lar)	* Uyumlaştırma * İyileştirme * Hızlı erişim * Karmaşık (kompleks) yapılar
İlişkisel (1970-80' ler)	* Daha çok güvenilirlik * Daha az tekrar (fazlalık) * Daha esnek * Çoklu görüntüler
Nesne Yönelimli (1990' lar)	* Daha iyi benzeşim (simulasyon) * Daha çok ve karmaşık veri tipleri * Daha fazla ilişki (Kümeleme, genelleme gibi) * Veri tabanı ve programlama için tek dil özelliği * Daha iyi uyarılma * Nesnelerin tekrar yapılandırılmaması * Tekrar kullanım, kalıtım gibi diğer avantajları

Nesneye yönelik veri tabanı yönetim sisteminde kullanıcı ve uygulamaların değişkenliği, yeni ihtiyaçları ortaya çıkarır. Bu nedenle bu ihtiyaçları karşılamak için sistem bir orta katı tanıtır. Bu katın ana amaçları, bir verinin istenilen forma ulaşması

⁷⁰ <http://www.dis.port.ac.uk/~chandler/OOlectures/database/database.htm>, 2002

için bir dönüşüm adımına izin vermekten çok, şeffaflık, yönetim ve değişim kolaylığı ile depolanan veride kavramsal olmayan sınırların karşılaştırılmasıdır. Şekil 11’de üç düzey mimarisinin kavramsal görünümü, kullanıcıların rolleri ve sorumluluklarıyla gösterilmiştir.⁷¹



Şekil 11: NYVTS’de üç düzey mimarisinin kavramsal görünümü.

2.3.2. NYVTS’nin Avantajları ve Dezavantajları

NYVTS’nin avantajları ve dezavantajları aşağıdaki gibi özetlenebilir:⁷²

NYVTYS’nin Avantajları:

- Nesnelere erişimin hızlanmasıyla tepe işlemlerde azalmaya neden olur ve her seferinde eleman tablolarından tekrar toplanmaya gerek duyulmaz. Bazı uygulamaların 100 defa daha hızlı olması gibi.
- Sayfalama azalmıştır.

⁷¹ http://www.pjwstk.edu.pl/~habela/publications/ADVIS2006Three-level_OODBA.pdf (Advanced i n Information Systems, 4th International Conference, ADVIS2006, Izmir, Turkey, October, 18-20 2006)

⁷² <http://www.dis.port.ac.uk/~chandler/OOlectures/database/database.htm>, 2002

- c) Uyarlama daha kolaydır.
- d) Veri tabanında dolaşma işaretleyici sayesinde daha kolay ve doğaldır.
- e) Tekrar kullanım geliştirme masraflarını azaltır.
- f) Uygunluk kontrolü, tam hiyerarşide tek kilit yeteneği ile basitleştirilmiştir.
- g) İlişkisel modele göre gerçek dünya işlemlerine daha iyi uyan bir veri modelidir.
- h) Veri tabanında anahtar bilgi içeren parçalar arasında başarılı uygulamalar yaratılabilir.
- i) Tek yerdeki değişiklikler sadece istemci uygulamalarında değil, sunucu uygulaması içinde de depolanmış olduğundan risk azalmıştır.
- j) İstemci/sunucu ve dağıtık mimarilerle uyumludur.

NVTYS'nin Dezavantajları:

- a) Uygulamalarda kullanılan anahtar bilgiyi içeren parçaların değeri azdır.
- b) Kalıtım hiyerarşisindeki geniş araştırmanın neden olduğu zaman kayıpları nedeniyle erişim hızı azalmıştır.
- c) Yapısal Sorgulama Dili (SQL) gibi bir genel sorgulama dili özelliklerine sahip değildir.
- d) NYVTS'de düzensiz anlamlar vardır. İlişkisel veri tabanında ise ilişkisel hesaplar grup aracı ile doğru bir şekilde ortaya konulabilir.
- e) İlişkisel tablolardaki basitlik kaybolmuştur.
- f) Nesne yönelimli paradigma değişikliklerini NYVTS'ne taşımak güçlükler yaratabilir.

2.4. Nesne Yönelimli Programlama Dilinin Özellikleri

Nesneye yönelik veri tabanı sisteminin nesne kimliği, karmaşık nesneler, türler ve sınıflar, sınıf hiyerarşisi, kalıtım gibi modelleme özellikleri yanında belirli dil özelliklerine de sahip olması gerekir.⁷³

⁷³ Yarımağan, Ü., a.g.m., s.212-215

- a) *Çok Biçimlilik (Polymorphism)*, “çok şekillik”olarak da isimlendirilir. Çok biçimlilik bir metotun farklı nesnelerde veya farklı ortamlarda farklı sonuçlar üretmesidir.

“Konuşmak” insan sınıfına ait metotdur. Ancak bu metot ait olduğu insan sınıfının örneğine göre farklı sonuçlar üretir. Bir Fransızın konuşması Fransızca olurken, bir İngilizin konuşması farklı seslerden oluşmaktadır. Konuşmak metodu farklı durumlarda farklı sonuçlar üretir, diğer bir ifade ile metodun birden fazla biçimi vardır⁷⁴.

- b) *Kapsülleme (encapsulation)*, nesnenin diğer nesnelere erişilebilir dış görünüşünün, diğer nesnelerden iç uygulama detaylarının saklanarak meydana getirilmesidir.⁷⁵ Veri tabanı bağlamında, şema düzeyinde, kapsülleme ile her sınıfın yapı bilgileri yanında davranış bilgilerinin de bulunduğu anlatılır. Davranış bilgisi, sınıftaki nesnelerin üzerinde uygulanabilen yöntemlerin (metotların) bu sınıfa atanabileceğini anlatır. Bu yöntemler dışarıdan *bir ileti* adı olarak görünür. Gerekiyorsa birkaç ileti argümanı ile sınırlıdır. Nesne bu iletilerden birini aldığı anda, bu iletiye karşı gelen ve sınıf düzeyinde tanımlanmış olan yöntemin uygulanmasını sağlar. Bu şekilde, bir sınıfa atanan bir yöntemin tanımında değişiklik yapıldığında, yöntemin dışarıdan görünen kısmı değişmeyeceği için kullanıcılar bu değişiklikten etkilenmeyecektir. Böylece mantıksal veri bağımsızlığı sağlanmış olacaktır.

Sınıf düzeninde bir yöntemin birden fazla tanımı yapılacağı için ileti adlarına birden çok anlam yüklenmiş olacaktır. Bir yöntemin diğerlerine göre öncelikli gerçekleşmesi üstünlük sağlama olarak bilinir. İleti adının üstünlük sağlamayı başarırken geç bağlama oluşur.

- c) *Diğer* bir özellik, küme-yönelik betimleyici dillere sahip olmaları ve kullanıcıların *anlık sorgularını* desteklemeleridir.

⁷⁴Zengin, A., Visial C#2005,Nirvana,Ankara, 2006, s.7

⁷⁵ Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., s.7

- d) *Nesne* yönelimli veri tabanlarında, yöntemin gerçekleştirilinde C++ ya da Smaltalk gibi bir dilin kullanılması durumunda, yöntemler sınıflara bağlanırken programlama dili ile bağlar kurularak veri tabanı dilinin hesaplama gücünün programlama dilinin *hesaplama gücü* düzeyine çıkarılması mümkün olmaktadır.
- e) *Sürüm yönetimi* ve *genişleyebilirlik* özelliğinin olması gerekmektedir. Ayrıca, NYVTS dillerinin *evrensellik*, *tanımlayıcılık*, *eniyilenebilirlik*, *kapalılık*, *yetkinlik*, *genellik*, *ifade gücü* gibi genel özelliklerinin bulunması gerekir.

Sorgu: Öğr-No = 117 olan öğrenci hangi dersi alır ?

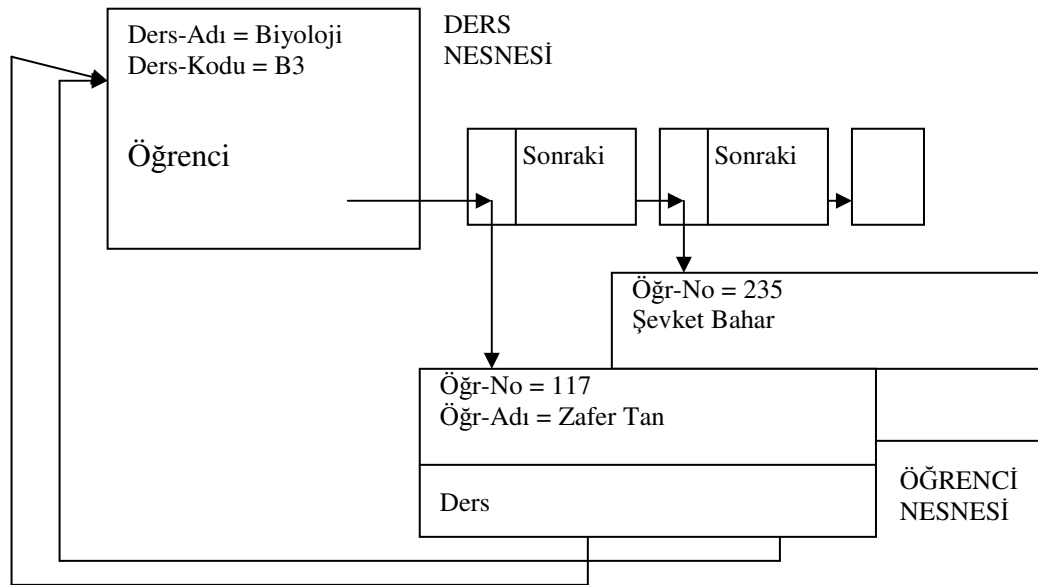
ÖĞRENCİ içinde 117 No'lu öğrenciyi belirler.

Ders-Kodu = B3'ün karşılığı olan Biyoloji'yi belirler.

Sorgu: Biyoloji dersini alan tüm öğrenciler kimlerdir ?

DERS içinde Ders-Kodu = B3 olan dersi bulur.

Ders-Kodu = B3 olan her Öğr-No'ya bakar.



Şekil 12: NYVM Öğrenci – Ders İlişkisi.

2.5. Nesne Yönelimli Modelleme Dili

Birleşik Modelleme Dili (Unified Modeling Language-UML)⁷⁶ iş modellemesi veya diğer sistemlerde olduğu gibi yazılım sistemlerinin tüm bölümlerini ayrıntılarıyla belirlemek, görüntülemek, resmetmek ve belgelemek için kullanılan standart bir dildir. Çok geniş ve karmaşık sistemlerin modellemesinde ve özellikle mühendis uygulamalarında başarılı şekilde kullanılan bir dildir. Bu dil, nesne yönelimli yazılım geliştirme sürecinin en önemli parçasıdır. Yazılım projelerinin tasarımında da grafiksel gösterimlerin yer aldığı bu dil sıklıkla kullanılır.

Amacı:

- a) Kullanıcılara geliştirilebilir ve değiştirilebilir, kullanmaya hazır, anlamsal gösterim modelleme yapabilmelerini sağlamak.
- b) Ana kavramları genişletmek için esneyebilir ve ayrıntıların belirtilebileceği mekanizma sağlamak.
- c) Belirli programlama dilleri ve geliştirme süreçlerini bağımsızlaştırmak.
- d) Modelleme dilinin biçimsel temelini oluşturmak.
- e) Nesne yönelimli araçlar piyasasının büyümesini teşvik etmek.
- f) İşbirliği, iskelet (çatı), kalıp (model) ve parça (eleman) gibi yüksek düzey geliştirme kavramalarını desteklemek.

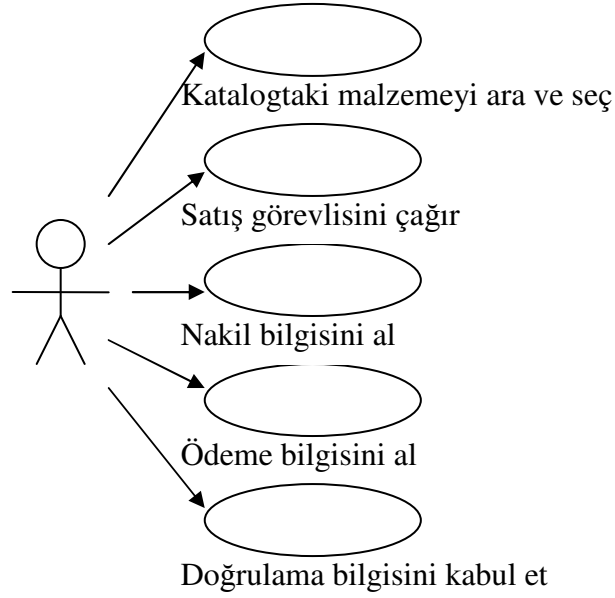
UML diyagram çeşitleri:

- a) *Kullanım biçimi diyagramı (Use case)* : Bu diyagramda bir kullanıcı ve sistem vardır. Bu ikisi arasındaki iletişim gösterilir. Örneğin, bir müşteri satış şirketine sipariş verdiğinde izlenen adımlar şu şekilde olacaktır.

- Katalogtaki malzemeyi ara ve seç.
- Satış görevlisini çağır.
- Nakil bilgisini al.
- Ödeme bilgisini al.

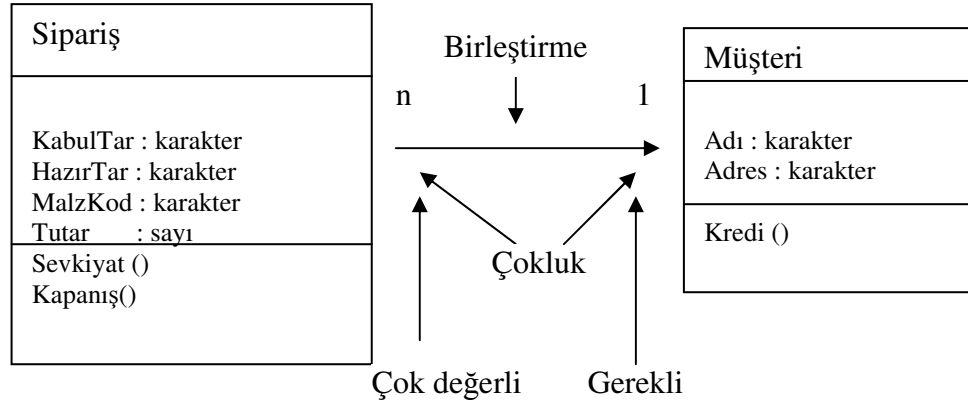
⁷⁶ http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/history_of_uml.htm

- Satıcıdan doğrulama bilgisini kabul et.



Şekil 13: Kullanıcı ve kullanım biçimleri

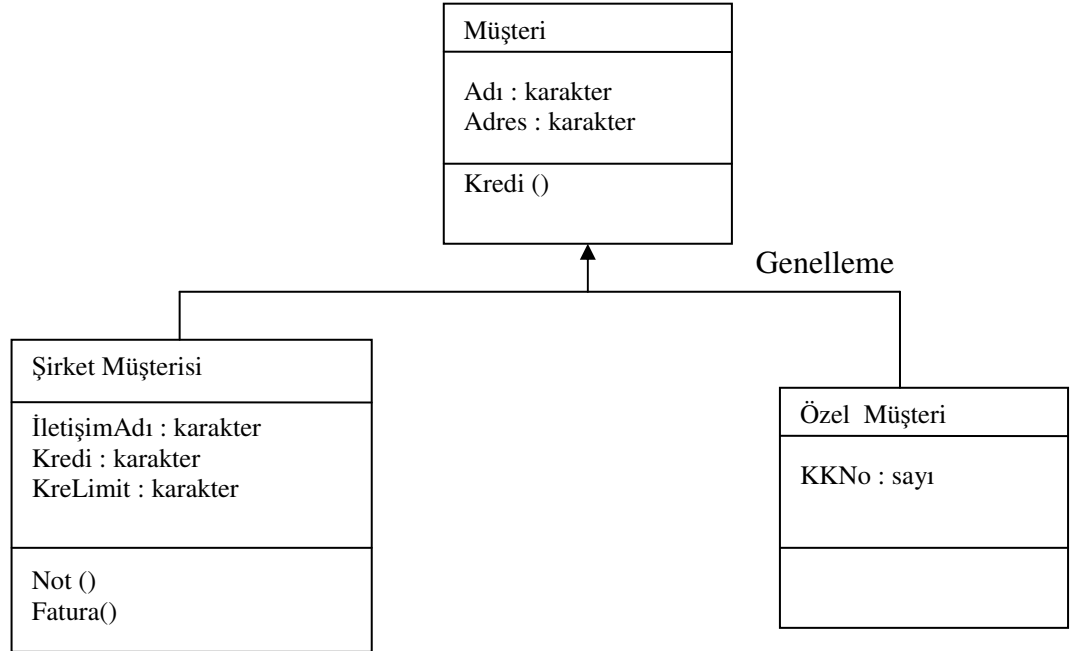
- b) *Sınıf diyagramları* : Sınıfta isim, nitelik, işlem olmak üzere üç özellik vardır. Kapsama, kalıtım, birleştirme ve diğer ilişkiler gösterilir.



Şekil 14: Birleştirme ilişkisi.

Sınıf diyagramında en genel ilişki birleştirme ilişkisidir. Birleştirme, sınıfların örnekleri arasındaki ilişkiyi gösterir. Örnekte, Müşteri sınıfı, Sipariş sınıfıyla birleşmiştir. Sipariş nesnesi, tek müşteri nesnesi ile birleşirken, bir müşteri nesnesi pek çok sipariş nesnesi ile birleşebilir.

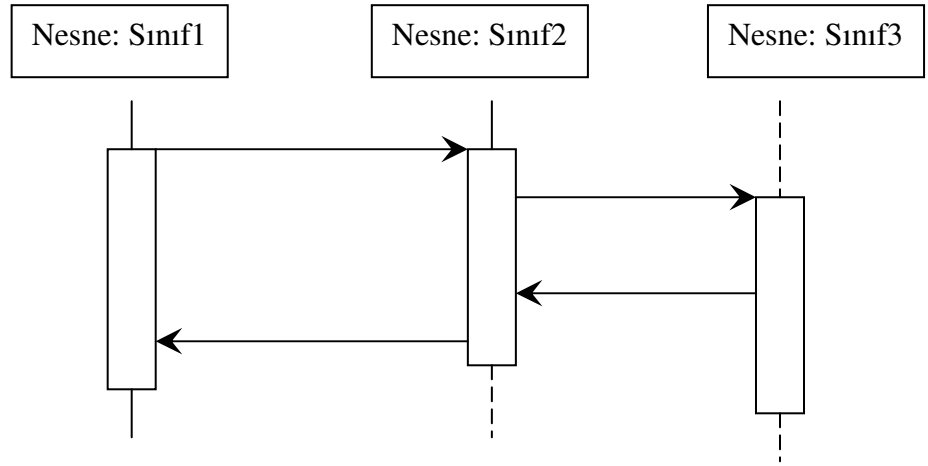
Diğer genel ilişki, genellemedir. Genelleme, benzer iki sınıf için yapılır ama bazı farklılıklar vardır.



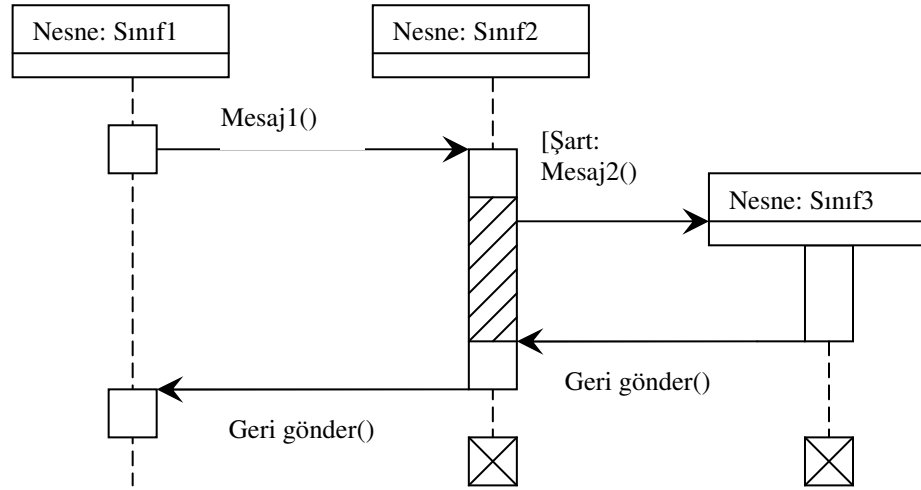
Şekil 15: Genelleme ilişkisi.

c) *Etkileşimli diyagramlar*: Nesne gruplarının tanımını ve davranışlarını gösterir. Sıra ve işbirliği diyagramları olmak üzere iki çeşittir.

- *Sıra diyagramı*: Örneğin sınıf1 nesnesi, sınıf2 nesnesine mesaj gönderme davranışını başlatır. Sonuç mesajı sınıf1 nesnesinde kabul edilinceye kadar farklı nesneler arasından geçer.



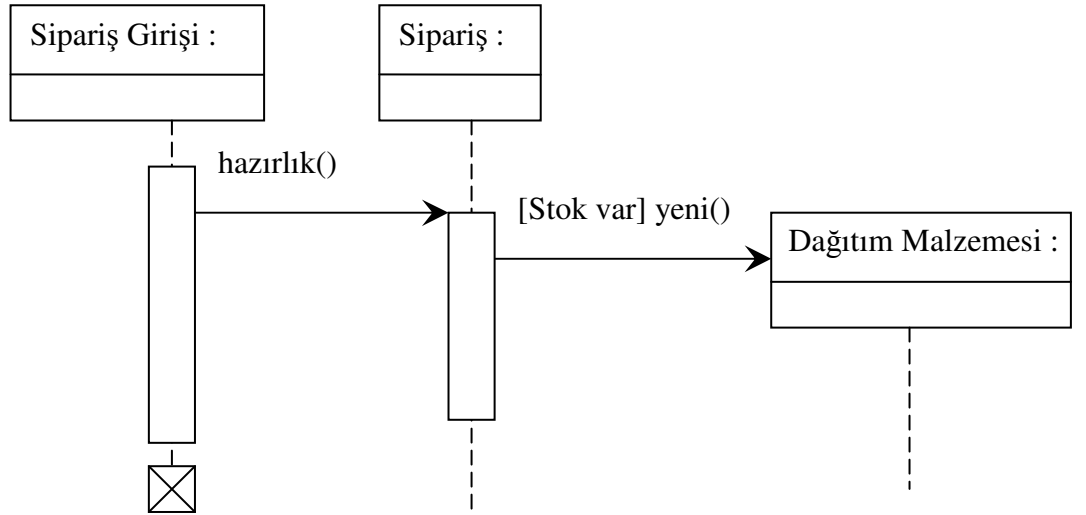
Şekil 16: Mesajın nesneler arasındaki davranışı.



Şekil 17: Mesajın nesneler arasındaki davranışı (şartlı).

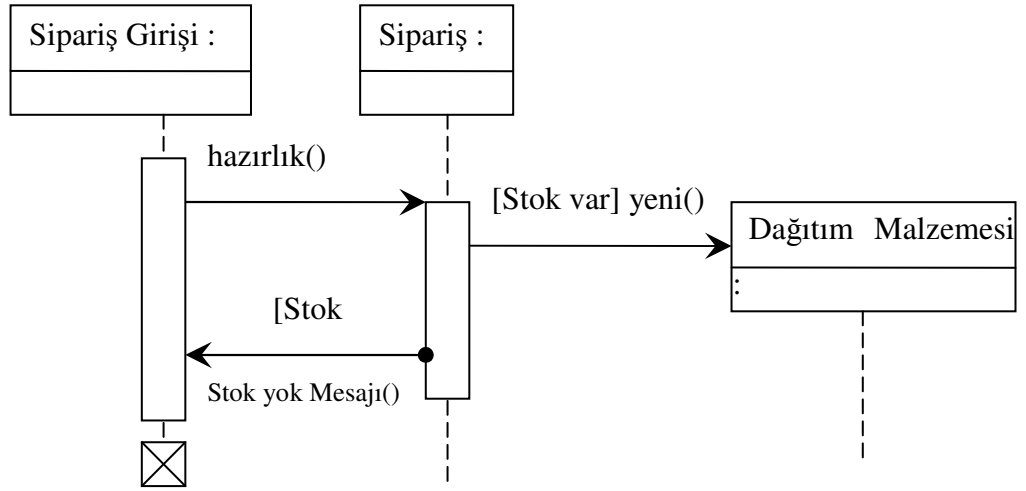
Durum daha karmaşık biçimde ortaya konulabilir: Sınıf3 nesnesine, sınıf2 nesnesinden mesaj gelmeden önce şart gösterilmiştir. Mesajın hareketi şarta göre oluşur.

Örnekte, sonraki diyagramlar bir siparişin yer aldığı sıra diyagramının başladığını gösterir. Sipariş girişi ile nesne yaratılır ve mesaj, sipariş hazırlığı için gönderilir. Nesne isimlerinin sütunlarla gösterildiğine dikkat edilmelidir. (Nesne-Adı:Sınıf-Adı).



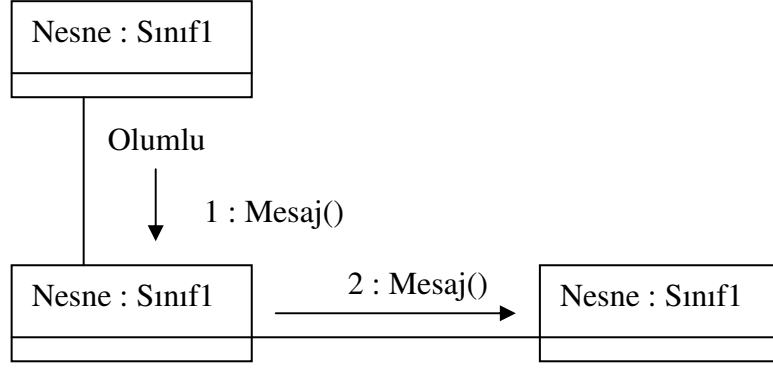
Şekil 18: Malzeme hazırlığı için mesajın hareketi.

Sipariş nesnesi kontrol edilir. Malzeme stokta varsa yeni bir malzeme yaratma mesajı gönderilir. Yoksa aşağıdaki gibi stokta yok mesajı gönderilir.

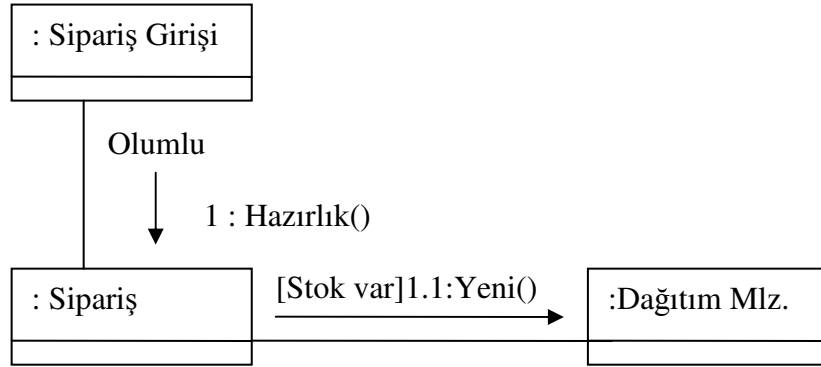


Şekil 19: Stokta varsa ya da yoksa durumuna göre mesajın hareketi.

- *İşbirliği diyagramı:* Bu diyagramlar nesneler arasındaki mesaj geçişlerini gösterir. Her mesaj bir sıra numarası ile belli edilir. 1,2,3,... Numaralandırması yapılır. Karmaşık diyagramlarda 1 için 1.1, 1.2., ... şeklinde 1.2 için 1.2.1, 1.2.2.,... şeklinde numaralandırma devam eder.

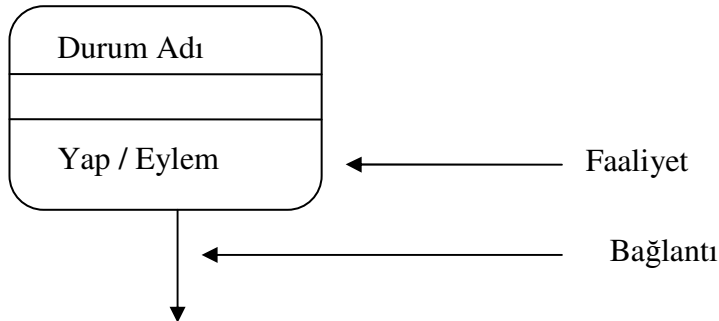


Şekil 20: Nesneler arası mesajın geçişi (İşbirliği diyagramı).



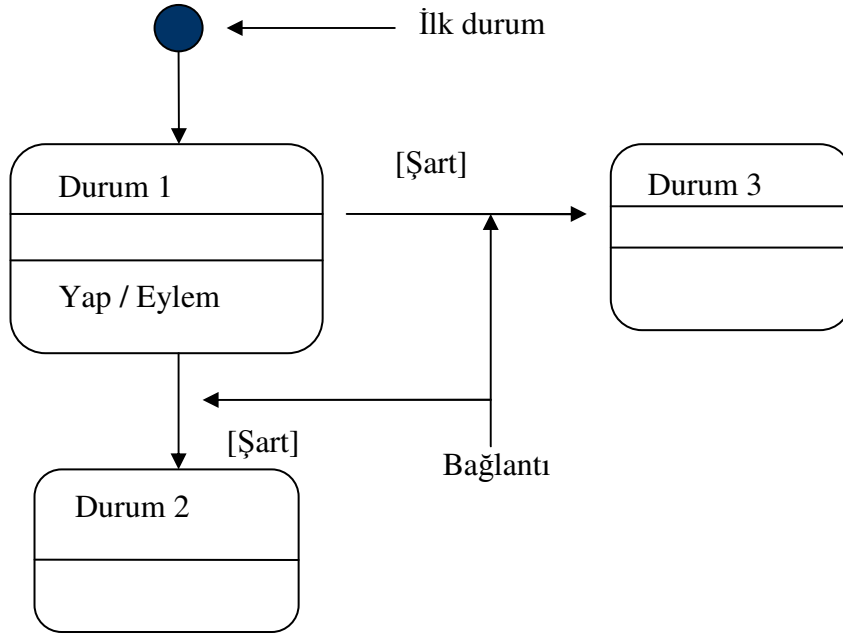
Şekil 21: Şarta göre malzeme hazırlığı (İşbirliği diyagramı).

- d) *Durum diyagramları* : Sistemin davranışlarını tanımlamak için kullanılır. Olayları meydana getiren nesnelerin mümkün durumlarının tüm tanımları yapılır. Her diyagram genellikle sistem boyunca nesnelerin farklı durumlarını, tek sınıf veya iz nesneleri olarak gösterir.



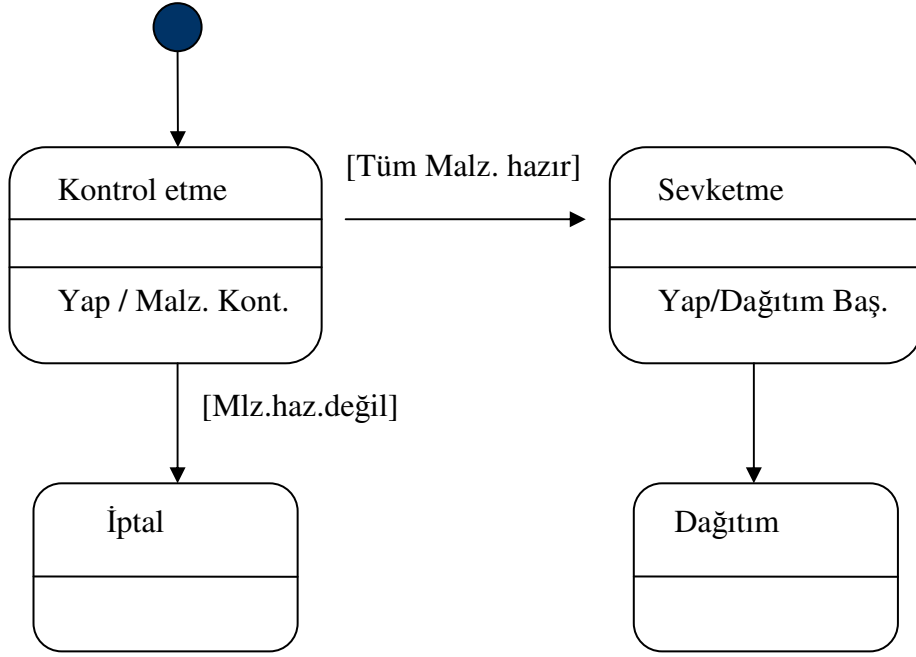
Şekil 22: Durum diyagramında nesnenin tanımı.

Bütün durum diyagramları nesnenin ilk durumuyla başlar. Bu durum yaratıldıktan sonra durumlar değişir. Şartlara göre faaliyetlere ve nesne durumlarına karar verilebilir.



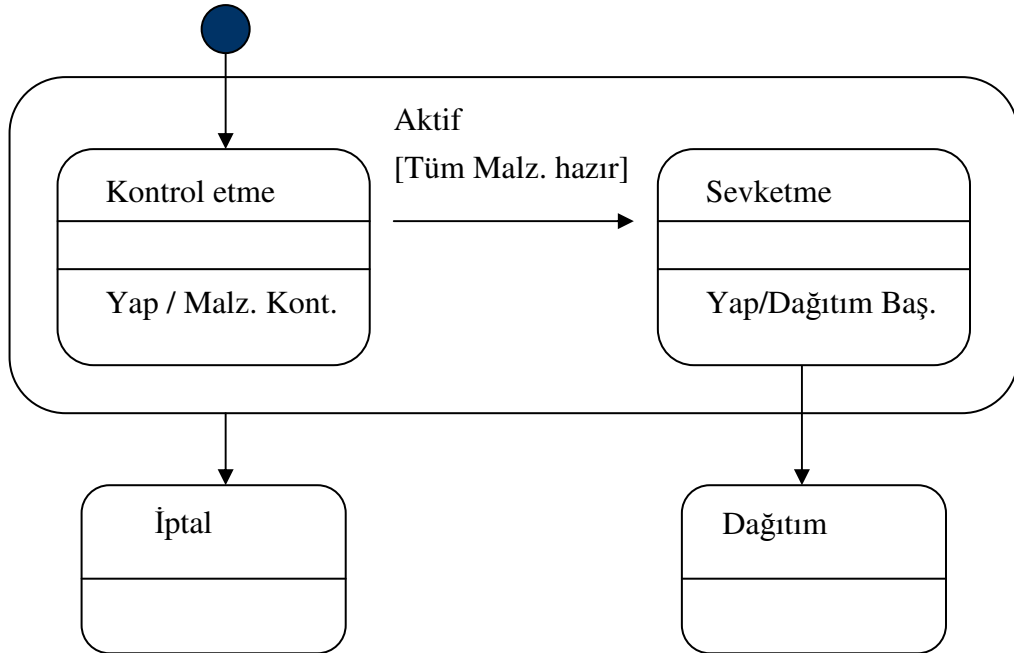
Şekil 23: Durum diyagramında ilk durum ve şartlar.

Aşağıda sipariş nesnesinin durum diyagramı görülmektedir. Nesne, kontrol durumunda girildiğinde malzeme kontrolü eylemini yapar. Eylem tamamlandıktan sonra şartlara göre durumlara bağlanır. Gereken eylem yapılır.



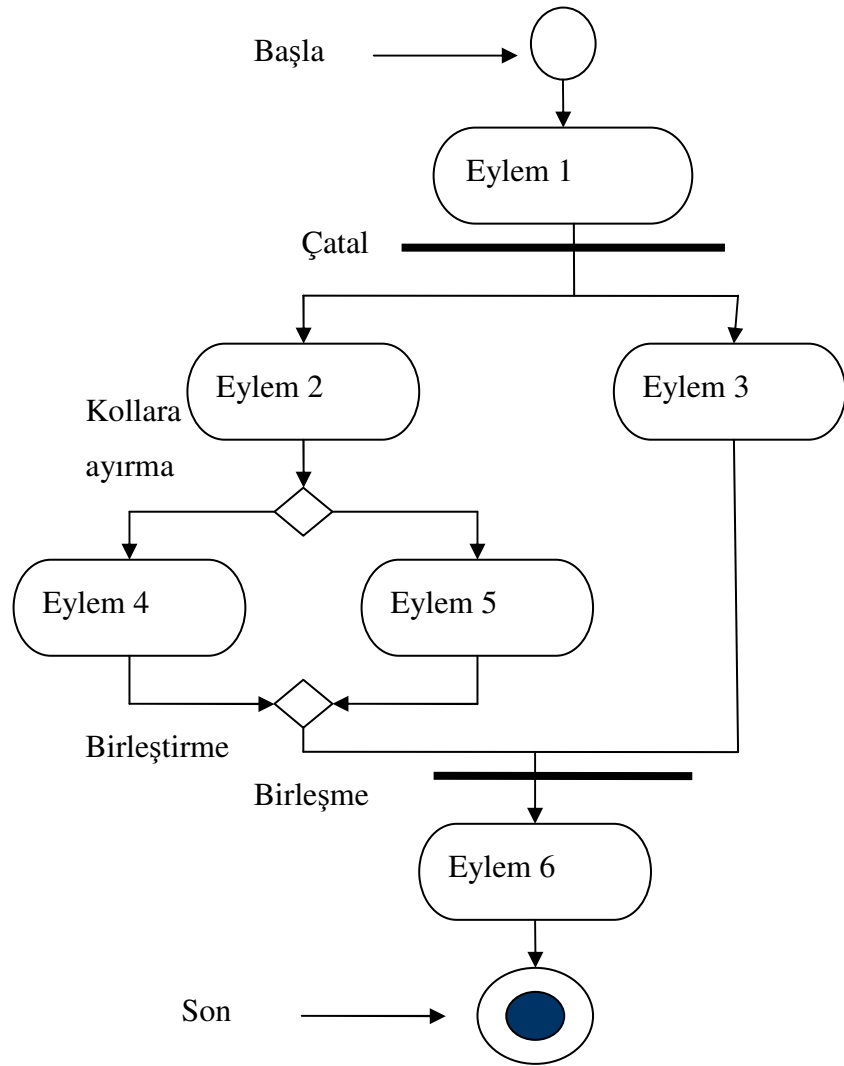
Şekil 24: Kontrol durumu.

Birden fazla durum aynı anda aktif olursa *süper durum* ortaya çıkar.

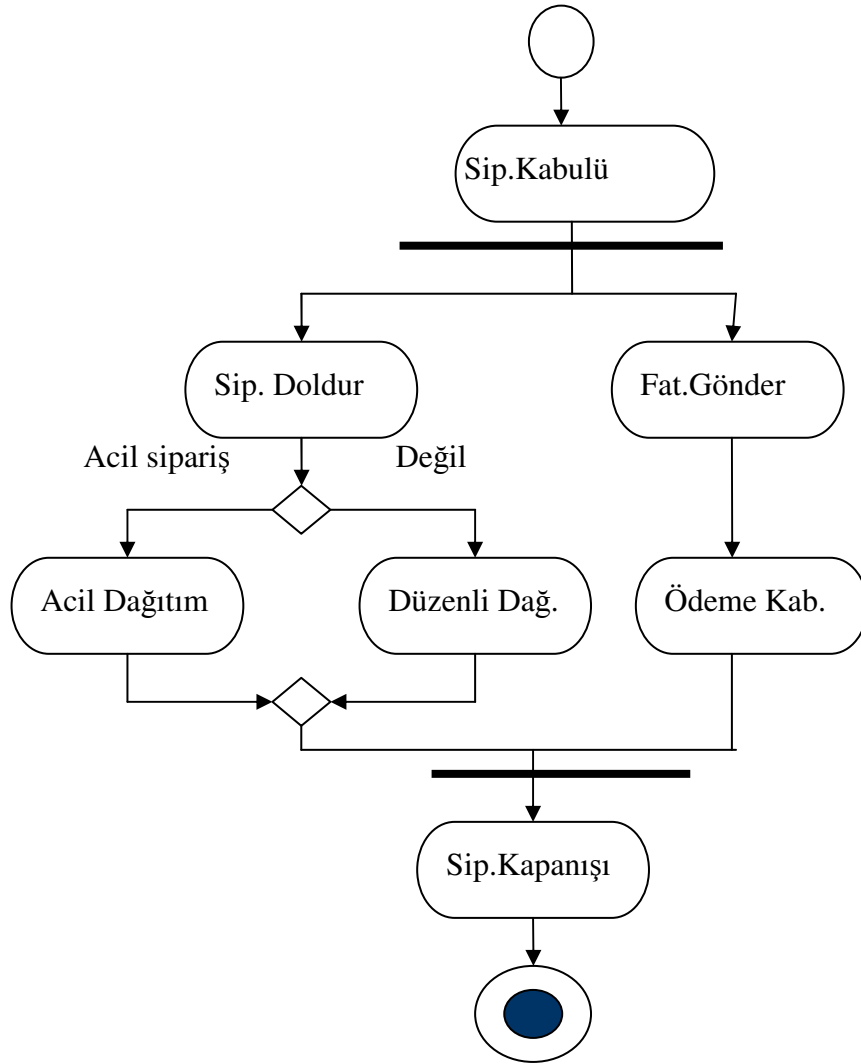


Şekil 25: Süper durum diyagramı.

- e) *Eylem diyagramları* : Sistem davranışlarını iş akışı olarak tanımlar. Durum diyagramlarına benzer. Çünkü eylemler, bir şeyin yapılması durumudur. Meydana gelen eylemlerin sırasını belirlemek için eylem diyagramları tanımlanır.

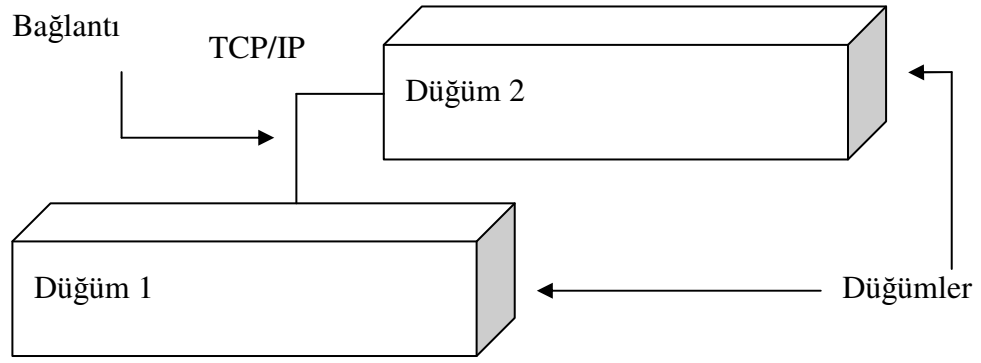


Şekil 26: Eylem diyagramı.

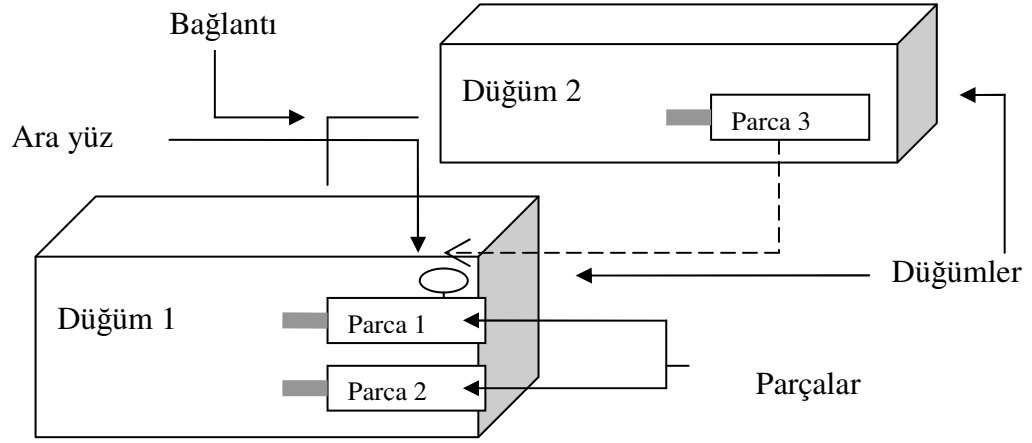


Şekil 27: Sipariş işlemi için eylem diyagramı.

- f) *Fiziksel diyagramlar:* Yayılma diyagramı ve parça diyagramı olarak iki çeşit fiziksel diyagram vardır. Yayılma diyagramı, sistemin donanım ve yazılımları arasındaki fiziksel ilişkiyi gösterir. Parça diyagramı ise, sistemin yazılım parçalarını ve aralarındaki ilişkiyi gösterir.



Şekil 28: Yayılma diyagramı.



Şekil 29: Bütün olarak fiziksel diyagram.

III. BÖLÜM

3. NESNEYE YÖNELİK VERİ TABANI UYGULAMASI

3.1. Sistem Analizi

Uygulama, Gümrük Müdürlüğü izni ile işletilen genel antrepoların yönetiminde kullanılmak üzere düşünülmüştür. Uygulamaya, depoda çalışacak ve bu uygulamaya malını Türkiye'ye gönderen firma, yurt içindeki alıcı ve gümrük kanunu ile ilgili prosedürleri yerine getirecek gümrük müşavirleri dahil olacaktır.

Bu uygulamada, depoya malın kabulü ve depo içi organizasyon yapılacaktır, alıcı tarafından üretim planlaması yapıldıktan sonra, uygulama üzerinden malın ithalatı ile ilgili işlemi verilecek, bu ithalat işleminin sonucu gümrük müşaviri tarafından uygulamaya girilecek ve depo personeli tarafından malın sevk edilmesi sağlanacaktır. O halde sistem analiz aşamasında uygulamanın sağlıklı biçimde gerçekleştirilebilmesi için konu ile ilgili yasal mevzuatın ortaya konulması gerekli olmuştur.

3.1.1. Gümrük

"Lehçe-i Osmaniye" adlı kitapta "Gümrük" kelimesi, Rumca'dan alınmış ve "emtiya ilişkin rüsumun idare mahallinin ismidir" diye tanımlanmış ise de, "Gümrük" kelimesinin Rumca'dan alınma olmayıp, Latince'de "ticaret" manasına gelen "Commercium" kelimesinden alınmış olduğu anlaşılmaktadır.

Gümrük resminin ticaret eşyasından alınan bir vergi olduğu gözönüne alındığında, kelimenin manaya uygunluğu ortaya çıkmaktadır. Fransızlar "Gümrük"e "Douane" derler. İtalyanca "Dogana" kelimesinden alınmıştır ki, Venedik ve Cenova Cumhuriyeti'nin Birinci Hakimi'nin ünvanı olan "Doge" adına, hazineye gelir temin etmek için, Venedik'te uygulamaya konulan verginin ismi olmuştur.

Venedikliler Ortaçağ'da ticarete büyük gelişme kaydettikleri zaman, bu rüsum uygulamaya konulmuş ve giderek "Gümrük" kelimesi, hem idarenin ismi (ism-i mekan) ve hem de verginin ismi (ism-i fiil) olarak kullanılmaya başlanılmıştır.

Zamanının ünlü yazarlarından Veyh'in sözlerine göre "Douane" kelimesi, gerek Farisi ve gerekse Arabi "Divan" kelimesinin aynısı olup, Ortaçağ'da Arap gümrüklerine gelen ticari emtiaya ilişkin rüsumun alınmasına memur edilen görevlilerin tamamı anlamına gelen ve "Divan" şeklinde adlandırılan bu ibarenin, o zamanki Latince'ye "Dogan"a şeklinde geçtiği ve oradan da Fransızca'ya "Douane" şeklinde geçerek, dillere yerleşmiş bulunduğu anlaşılmaktadır. (Süleyman Sudi/Mehmet Ali Ünal (Osmanlı Vergi Düzeni/Defter-i Muktesid))⁷⁷

3.1.2. Eşya İthalatı

İthal edilen eşyalar ile ilgili Gümrük Müdürlüğü yönetmeliğinin ilgili kanun ve maddeleri aşağıdaki gibidir.⁷⁸

Geçici Depolanan Eşya:

Madde 73 : Türkiye Gümrük Bölgesine getirilen serbest dolaşımda olmayan eşya gümrüğe sunulmasından sonra gümrükçe onaylanmış bir işlem veya kullanıma tabi tutuluncaya kadar geçici depolanan eşya statüsünde bulunur ve bu şekilde adlandırılır. Talep halinde ihracat eşyasının da bu kapsamda değerlendirilmesi mümkündür.

Geçici Depolama Yerleri:

Madde 74 : Geçici depolanan eşyanın her türlü dış etken ve müdahalelerden korunmasını sağlayacak şekilde yapılmış ve taşıtların durduğu, yanaştığı veya indiği yerlere mümkün mertebe yakın olan ambar, depo, ardiye veya hangar gibi yerler geçici depolama yerleridir.

⁷⁷ http://www.gumruk.gov.tr/Content.aspx?cT=0&cId=0_0_0_0

⁷⁸ 4458 sayılı Gümrük Kanunu

Gümrük Antrepo Rejimi:

Tanımlar

Madde 277 : Bu rejimin uygulanmasında;

- a) *Gümrük antreposu*; gümrük gözetimi altında bulunan eşyanın veya izin verildiği durumlarda ihraç eşyasının konulduğu genel ve özel antrepoları,
 - b) *Genel antrepo*; eşyanın konulması için herkes tarafından kullanılabilen gümrük antrepolarını,
 - c) *Özel antrepo*; sadece antrepo işleticisine ait eşyanın konulabildiği gümrük antrepolarını,
 - d) *İşletici*; gümrük antreposu işletme izni verilen kişiyi,
 - e) *Kullanıcı*; eşyanın antrepo rejimi beyanında bulunan kişi veya bu kişinin hak ve yükümlülüklerinin devredildiği kişiyi,
 - f) *İhracata bağlı önlemlerden yararlanan eşya veya ürün*; ihracı halinde bir geri ödeme veya benzeri ekonomik fayda sağlayan eşya veya ürünleri,
- ifade eder.

3.1.3. Taşıtların Türkiye Gümrük Bölgesine Giriş ve Çıkışı

Madde 33 : Türkiye Gümrük Bölgesine giriş ve çıkış gümrük kapılarından yapılır. Türkiye Gümrük Bölgesinin giriş noktalarındaki gümrük kapıları ile içeride bulunan gümrük kapıları arasında belirli yolların takip edilmesi zorunludur. Giriş ve çıkış kapıları ile bunları birbirine bağlayan yollar ve hava taşıtlarının Türkiye Gümrük Bölgesinde inebilecekleri gümrük işlemi yapılan havalimanları ilgili kamu kuruluşlarının görüşleri alınarak, Müsteşarlıkça tespit edilerek Resmi Gazetede yayımlanır.

Genel hizmete açık demiryolları gümrük yolu sayılır.

Madde 34 :

1. Türkiye Gümrük Bölgesine giren veya çıkan taşıtlar gümrük gözetimine tabidir. Bunlar, yürürlükteki hükümlere uygun olarak, gümrük idareleri tarafından denetlenir.

2. Karayolu ile Türkiye Gümrük Bölgesine gelen taşıtların denetlenmesi bitmeden veya ilgili gümrük idaresinin izni alınmadan söz konusu taşıtlara yük ve yolcu alınıp verilemez ve taşıt yoluna devam edemez; trenlerin vagon değiştirmek veya eklemek suretiyle tertipleri değiştirilemez.

3.1.4. Eşyanın Türkiye Gümrük Bölgesine Girmesi

Madde 36 :

1. Türkiye Gümrük Bölgesine getirilen eşya, girişinden itibaren gümrük gözetimine tabidir. Bunlar, yürürlükteki hükümlere uygun olarak gümrük idareleri tarafından denetlenir.

2. Söz konusu eşya gümrük statüleri belirleninceye, serbest dolaşımda olmayan eşya ise 77 nci maddenin 1 inci fıkrası hükmü saklı kalmak üzere, gümrük statüleri değişinceye ya da serbest bölgeye girinceye yahut 163 ve 164 üncü maddeler gereğince yeniden ihraç veya imha edilinceye kadar gümrüğün gözetimi altında kalır.

Madde 37 :

1. Türkiye Gümrük Bölgesine getirilen eşya, getiren kişi tarafından gecikmeksizin Müsteşarlıkça belirlenen usul ve esaslara uygun olarak;

a) Belirlenen bir gümrük idaresine veya gümrükçe uygun görülen herhangi bir yere,

b) Deniz veya havayoluyla ya da Türkiye Gümrük Bölgesinden geçmeksizin karayoluyla doğrudan bir serbest bölgeye, götürülür.

2. Türkiye Gümrük Bölgesine getirildikten sonra eşyanın aktarılması sonucunda nakliyesinden sorumlu olanlar yukarıda belirtilen hükümlere uymak zorundadır.

3.1.5. Eşyanın Gümrüğe Sunulması

Madde 39 : Müsteşarlıkça belirlenen esaslara uygun olarak serbest bölgelere konulan eşya hariç olmak üzere, gümrük idaresine veya gümrük idarelerinin belirlediği veya uygun gördüğü diğer bir yere gelen eşya, bunu Türkiye Gümrük Bölgesine getiren kişi veya yerine göre eşyanın gelişinden sonra taşımasını üstlenen kişi tarafından gümrüğe sunulur.

Madde 40 : Müsteşarlık;

- a) Yolcu beraberinde getirilecek,
 - b) Gümrüğe sunulmaksızın bir gümrük rejimine tabi tutulabilecek,
- Eşya hakkında, 39 uncu madde hükümleri dışında özel düzenlemeler yapılabilir.

Madde 41 : Talep üzerine, gümrükçe onaylanmış bir işlem veya kullanıma tabi tutulmasından önce, gümrüğe sunulan eşyanın incelenmesi veya bundan numune alınması yönünden ilgili ve yetkili kişilere izin verilebilir.

3.1.6. Özet Beyan ve Gümrüğe Sunulan Eşyanın Boşaltılması

Madde 42 : 44 üncü madde hükümleri saklı kalmak üzere, 39 uncu madde uyarınca gümrüğe sunulan eşya için, eşyanın gümrüğe sunulmasını takip eden ilk iş günü mesai bitimine kadar ilgili gümrük idaresine bir özet beyan verilir.

Madde 43 :

- a) Özet beyan, Müsteşarlıkça belirlenen örneğe uygun bir form ile yapılır. Müsteşarlık, eşyanın teşhisi için gerekli ayrıntıları içeren ve uluslararası kabul gören bir ticari veya resmi belgenin de özet beyan olarak kullanımına izin vermeye yetkilidir.
- b) Özet beyan;
 - Eşyayı Türkiye Gümrük Bölgesine getiren veya eşyanın gelişinden sonra taşıma sorumluluğunu üstlenen,

- (a) bendinde belirtilenlerin adına hareket eden,
Kişi tarafından verilir.

Madde 44 : Yolcu beraberinde getirilen eşya ile yolculara ait olmakla birlikte başka taşıtlarla getirilen eşya ve posta yoluyla gönderilen koliler için özet beyan aranmaması hakkında özel hükümler koymaya Müsteşarlık yetkilidir.

Ancak, özet beyan aranmayacak hallerde, 42 nci maddede belirtilen sürenin dolmasından önce, eşyanın gümrükçe onaylanmış bir işlem veya kullanıma tabi tutulması ve gümrük denetiminin tehlikeye düşmemesi şarttır.

Madde 45 :

- a) Eşya bulunduğu taşıt araçlarından gümrük idarelerinin belirledikleri veya uygun gördükleri yerlerde söz konusu idarelerin izni ile boşaltılabilir veya aktarılabilir. Gümrük idaresine özet beyan veya özet beyan olarak kullanılan ticari ya da resmi belge verilmeksizin taşıtlardan eşya boşaltılamaz.

Ancak, eşyanın tamamen veya kısmen acilen boşaltılmasını gerektiren kaçınılmaz bir tehlikenin varlığı durumunda bu izin aranmaz. Bu gibi durumlarda en yakın gümrük idaresi derhal haberdar edilir.

- b) Gümrük idareleri, eşyanın ve bulunduğu taşıma araçlarının muayenesi amacıyla gerektiğinde eşyanın boşaltılmasını ve kapların açılmasını isteyebilir.
- c) Eşya konulduğu ilk yerden gümrük idarelerinin izni olmaksızın kaldırılamaz.

3.1.7. Gümrüğe Sunulan Eşyaya Gümrükçe Onaylanmış Bir İşlem veya Kullanım Belirlenmesi Zorunluluğu

Madde 46 :

- a) Gümrüğe sunulan eşyaya, gümrükçe onaylanmış bir işlem veya kullanım tayin edilir.

b) Özet beyan kapsamındaki eşyaya, gümrükçe onaylanmış bir işlem veya kullanım belirlenerek, buna ilişkin işlemler;

- Denizyolu ile gelen eşya için, özet beyan verildiği tarihten itibaren kırkbeş gün,
- Diğer bir yolla gelen eşya için, özet beyanın verildiği tarihten itibaren yirmi gün, içinde tamamlanır.

c) Şartlar gerektirdiği takdirde, Müsteşarlık 2 nci fıkrada belirtilen sürelerden daha kısa bir süre saptayabilir veya bu sürelerin uzatılmasına izin verebilir. Ancak, gerçek ihtiyaçları aşan süre uzatımı yapılamaz.

3.1.8. Eşyanın Geçici Depolanması:

Madde 47 : Eşya, gümrüğe sunulmasından sonra gümrükçe onaylanmış bir işlem veya kullanıma tabi tutuluncaya kadar geçici depolanan eşya statüsünde bulunur ve bu şekilde adlandırılır.

Madde 48 :

1. Geçici depolanan eşya, sadece gümrük idarelerinin uygun gördüğü yerlerde ve bu idarelerin belirlediği koşullarda depolanabilir.

Gümrük idareleri, geçici depolanan eşya için tahakkuk edebilecek gümrük vergilerinin ödenmesini sağlamak üzere eşya sahibinden teminat isteyebilir.

2. Yolcu beraberinde getirilip gümrüğe sunulmasından sonra gümrükçe onaylanmış bir işlem veya kullanıma tabi tutuluncaya kadar yolcu eşyasına mahsus gümrük ambarlarına konulan eşyanın buralarda kalabileceği süre üç aydır.

Madde 49 : 41 inci madde hükümleri saklı kalmak üzere, geçici depolanan eşya, görünüş ve teknik özelliklerinin değiştirilmemesi koşuluyla, aynı durumda muhafazalarını sağlamak üzere yönetmelikle belirlenen elleçlemeye tabi tutulabilir.

Madde 50 :

- a) 46 ncı maddeye göre belirlenmiş süreler ile 48 inci maddenin 2 nci fıkrasında belirtilen süre içinde kendilerine gümrükçe onaylanmış bir işlem veya kullanım tayini için gerekli işlemlere başlanmamış eşya, herhangi bir adli veya idari takibata konu olmaması halinde, 177 ila 180 inci madde hükümlerine göre tasfiye edilir.
- b) Gümrük idareleri, söz konusu eşyayı durumu belirleninceye kadar, eşya sahibinin risk ve hesabına gümrüğün denetimindeki özel bir yere sevk edebilir.

3.2. Genel Antrepo Yönetim Sistemi Etkileşimli Birimleri

Genel Antrepo Yönetim Sistemi uygulamamıza dahil olacak birimler aşağıda belirtilmiştir.

1. Depo:

İthal edilmek istenen eşyaların serbest dolaşım izni çıkana kadar geçen sürede muhafaza edildiği genel antrepolardır. Yurt içinde Gümrük Bakanlığı'nın verdiği izin dahilinde işlem yaparlar. Mal kabul ve sevkiyat işlemi, Gümrük Bakanlığı'nın görevli gümrük memuru tarafından denetlenir.

2. Alıcı:

Yurtdışından hammadde ya da yarı mamül ithalatını yapan firmalardır. Bu firmalar bir ürünün dağıtıcısı ya da üretim yapan işletmelerdir.

3. Tedarikci:

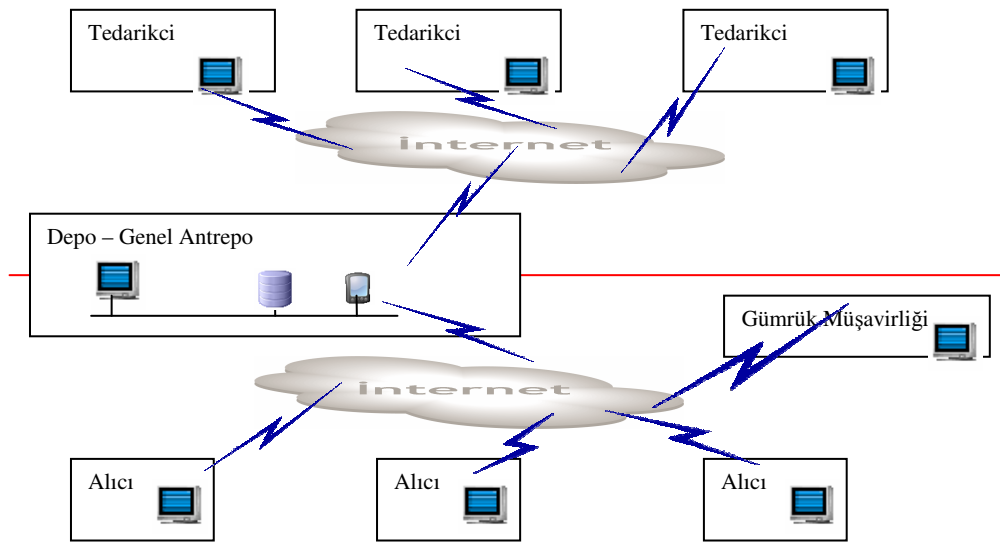
Buradaki Tedarikci terimi; üretim yapan işletmelere ya da yurt içindeki dağıtıcılara hammadde ya da yarı mamül teminini gerçekleştiren yurtdışındaki firmalardır.

Tedarik edilen ürünler konsinye ya da direkt satış yöntemi ile gönderilir. Konsinye ürünler ön fatura ile antrepoya alınır. İthalat işleminden önce gerçek satış işlemi gerçekleştirilerek faturası oluşturulur.

4. Gümrük Müşavirliği:

İthalat işleminde; alıcı firmanın vekaleti ile alıcı firmaya müşavirlik hizmeti veren yetkili firmalardır. Antrepoya alma ve ithalat işlemi, A tipi gümrük karnesine sahip müşavirler tarafından hizmet verilerek gerçekleştirilir.

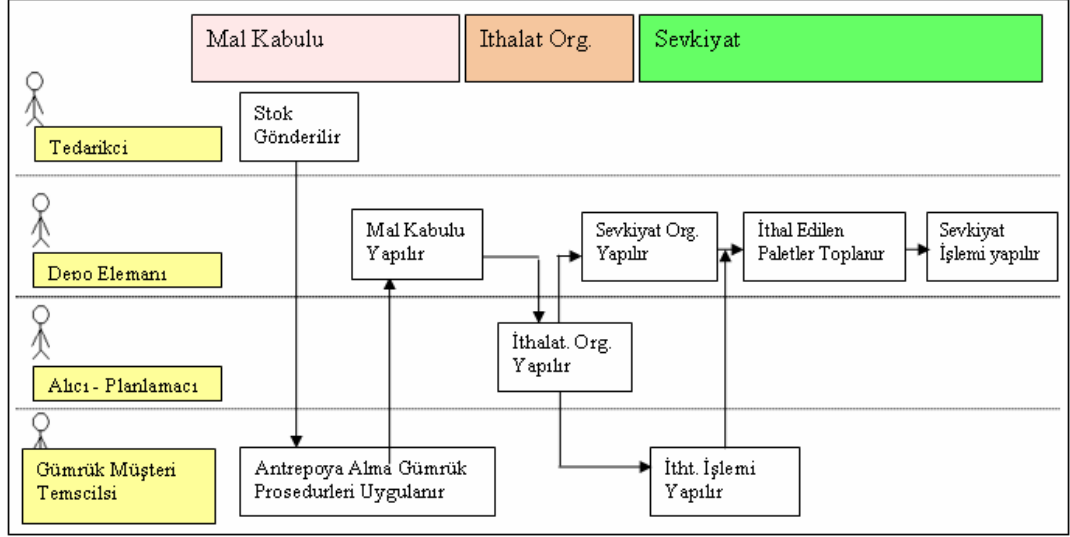
3.3. Genel Antrepo Yönetim Sisteminin Şemaları



Şekil 30: Veri iletişim yolu.

Uygulama 3 katmanlı mimariye sahiptir. Sunucu ve veri tabanı depoda bulunmaktadır.

Sunucuya depo personeli yerel ağdan; tedarikçi, alıcı ve gümrük müşaviri internet üzerinden ulaşır.

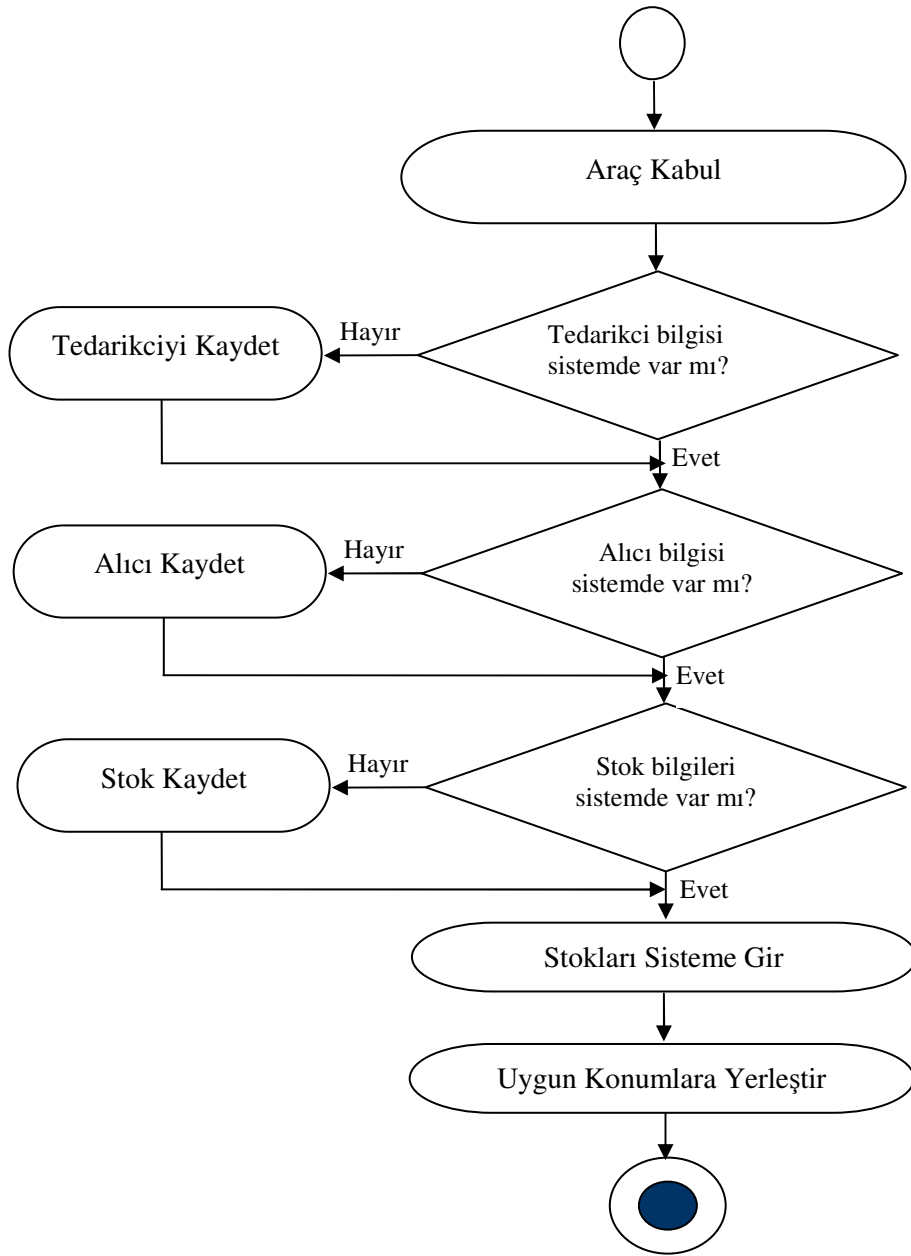


Şekil 31: Fonksiyonlar arası süreç haritası.

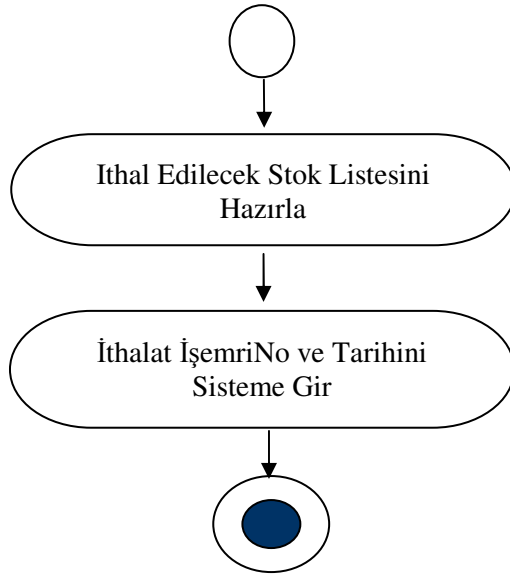
Şekil 31'deki süreç haritası incelendiğinde ilk süreç mal kabulü sürecidir. Tedarikçi ve alıcının kendi sistemlerinde takip ettikleri siparişlere istinaden, tedarikçi tarafından stoklar gönderilir. Gümrük müşaviri, devreye girerek antrepoya alma sürecini başlatır. Antrepoya alma işlemindeki yasal süreç gerçekleştikten sonra, stoklar depo personeli tarafından depoya alınarak uygun konumlara yerleştirilir.

Mal kabulü gerçekleştikten sonra ithalat organizasyon süreci başlar. Alıcı; yaptığı üretim planlamasına göre hangi stoğun hangi tarihte ithal edileceğini stok kayıtları üzerine girer. İthalat işlemi başlayana kadar bu işlemi bilgilerini değiştirme hakkına sahiptir.

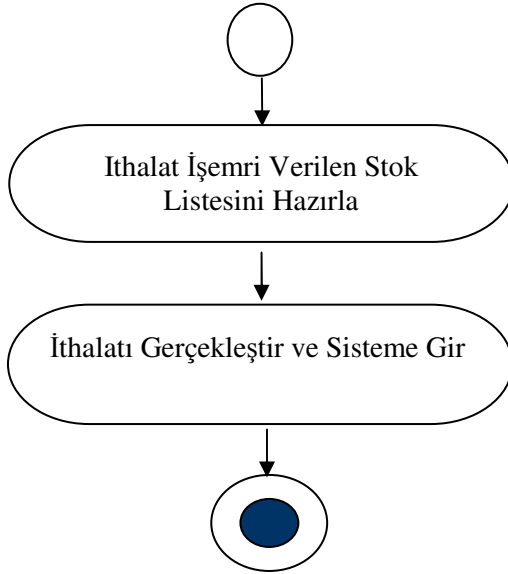
İthalat organizasyonundan sonraki süreç, sevkiyat sürecidir. Bu süreçte depo personeli, o gün için ithalat yapılacak stokları görerek sevkiyat organizasyonunu başlatır. Aynı zamanda gümrük müşaviri ithalatı gerçekleştirir ve ithalat işlemi bittikten sonra ithalatı gerçekleşen stoklara Gümrük Müdürlüğü'nün verdiği ithalat numarasını girerek stokların sevk edilebileceğini sistem üzerinden iletir. İthalat işleminin tamamlandığını gören depo personeli, stokları araca yükleyerek sistemden çıkışı sağlar.



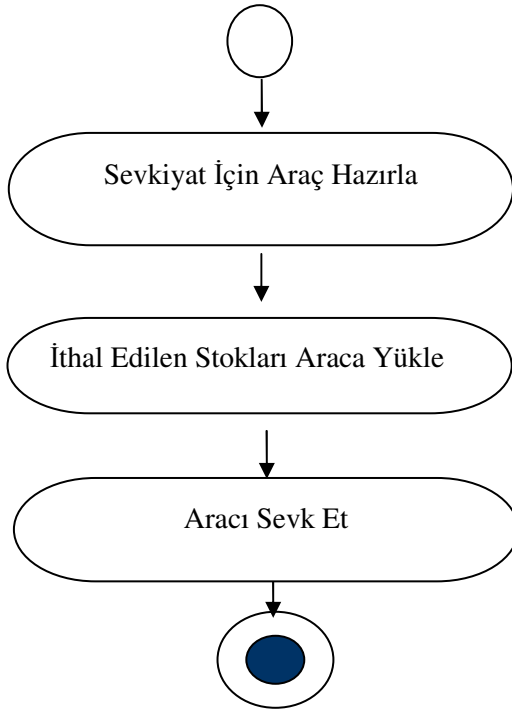
Şekil 32: Mal kabülü eylem diyagramı.



Şekil 33: İthalat organizasyon eylem diyagramı.



Şekil 34: İthalat eylem diyagramı.



Şekil 35: Sevkiyat eylem diyagramı.

3.4. Sistem Analizinden Elde Edilen Sonuç

Analiz incelendiğinde etkileşimli birimler arasındaki iletişimin büyük çoğunluğunun sistem üzerine çekilerek zaman ve işgücü tasarrufu yapılması sağlanmış, hata riski en az seviyeye indirilmiştir.

Depo personeli gelecekteki iş yoğunluğunu görececek, personel ve araç planlaması yaparak, gereksiz iş kayıplarını önleyecektir. Stoklara atanmış ithalat işlemi tarihine dayanarak, stokların sevkiyat sıralarını bilecek, konum organizasyonunu yapacak, ilk sevk edilecek stokları ulaşılması kolay konumlara koyacaktır. Ayrıca depoda birden fazla alıcının stokları olduğu düşünülürse, stokları alıcı bazında da gruplayabilecektir.

Gümrük Müşaviri de işemirlerini sistemden görerek gerekli evrakların hazırlanmasını planlayacaktır.

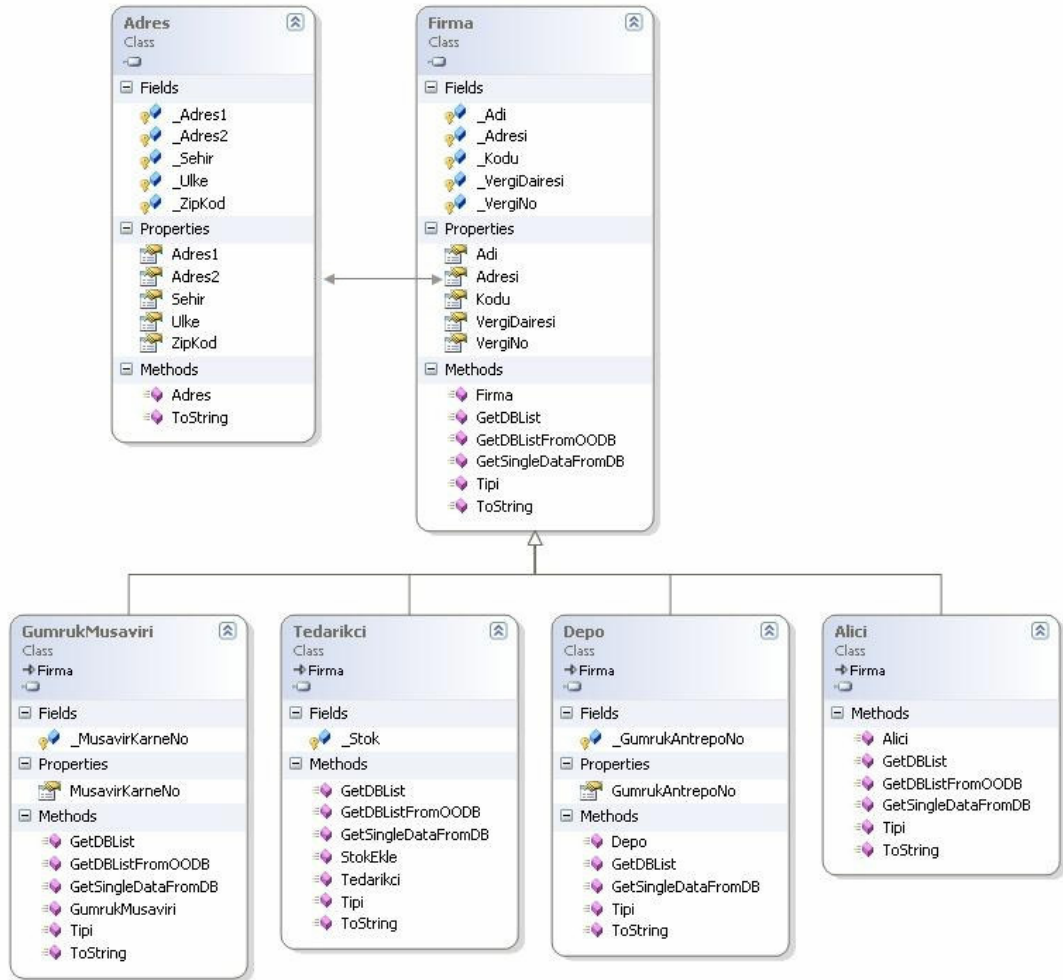
Depo, g mr k m  aviri ve alıcı, aynı stok bilgilerini g rd    i in genelde g mr k m  aviri i in bilinmesi zor olan stok kodunun bilinmesine ihtiya  kalmayacak ve ileti imde ya anan problemlerden dolayı istenenin dı ında stoklar i in i lem yapılması  nlenecektir.

Alıcı tarafından sistem incelendi inde,  retim planlamasını sistem  zerinde tutarak hata riskini azaltacak, sistem g venirlili i arttık a  retim alanına fazla stok  ekmeyerek  retim alanını daha verimli kullanacaktır.  retim i in gerekli stokların zamanında ve do ru gelmesi  retim kapasitesini arttıracaktır.

Tedarikci a ıcından sistem incelendi inde, m  terisi i in g nderdi i stokların durumunu takip ederek ithal edilen stokların yerine yenilerini g nderecek, m  terisinin g venlik stok de erlerinin altına inmeyecektir.

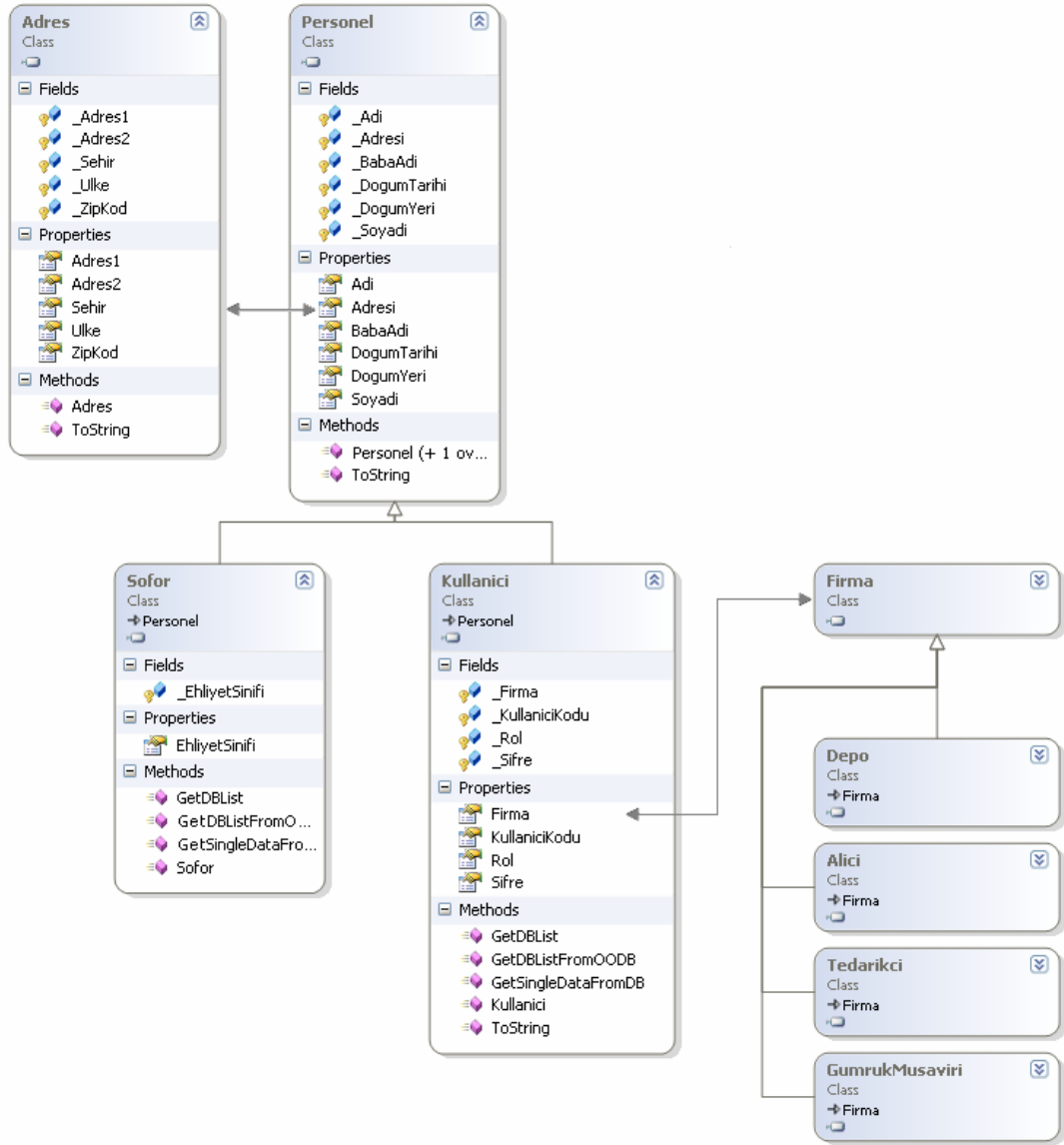
3.5. Sistem Tasarımı

Sistem analizi  alı masına dayanarak NYVT'ne dayalı yeni bir sistem tasarımı yapılmı tır. Uygulamada gerekli olan sınıflar yaratılarak aralarındaki ili ki diyagramlarda g sterilmi tir.



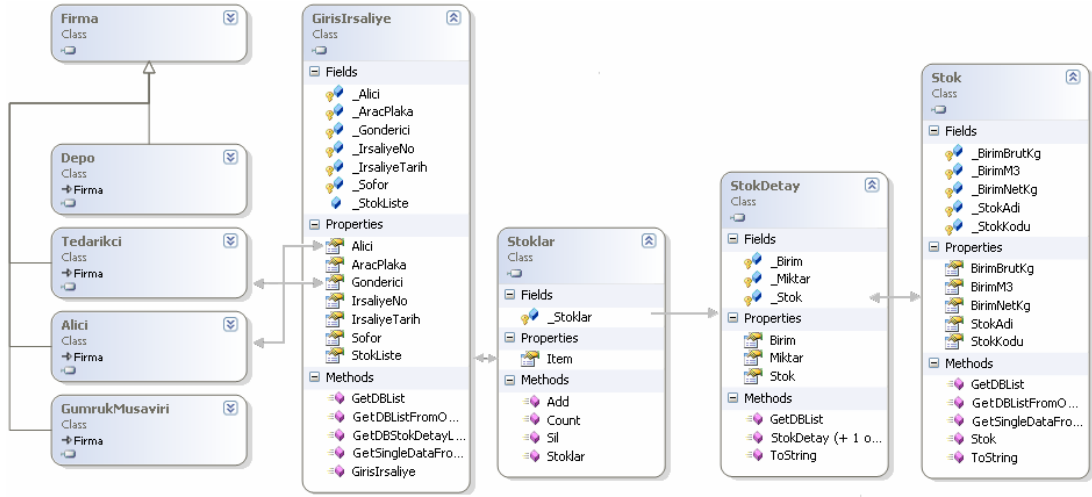
Şekil 36: Firma sınıf diyagramı.

“Adres sınıfı”, “Firma sınıfı”nın bir alt sınıfıdır. Aralarında bire-bir ilişki vardır. “Adres sınıfı”ndan “Gümrük Müşaviri”, “Tedarikçi”, “Depo”, “Alıcı sınıf”ları türetilmiştir. Türetilen bu sınıflar “Firma sınıfı”nın tüm özellik ve metotlarını miras almışlardır ve kendilerine özgü nitelikleri vardır.



Şekil 37: Personel sınıf diyagramı.

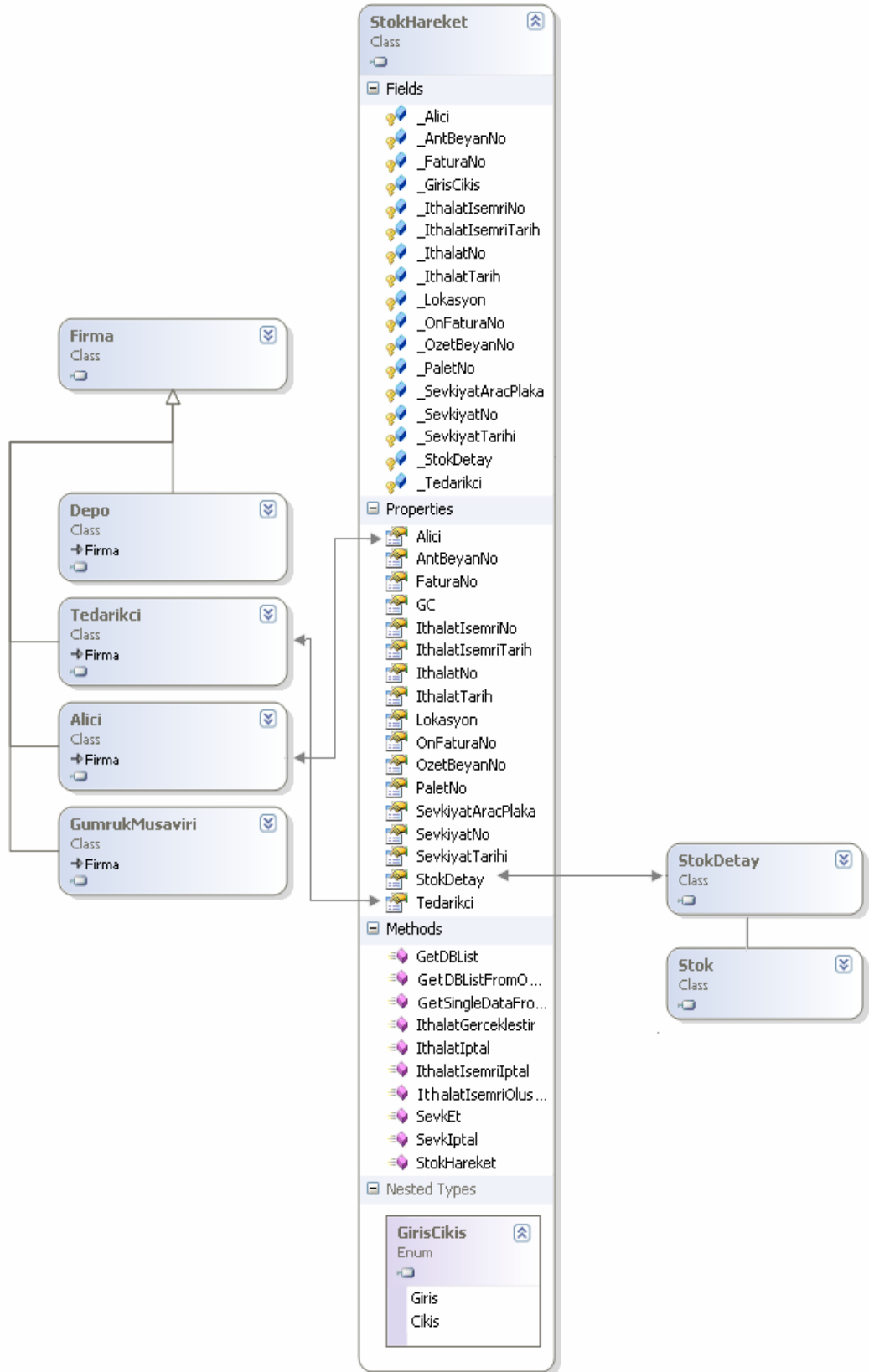
“Personel sınıfı” kendisinden başka sınıflar türetilmek için yaratılmıştır. “Personel Sınıfı”ndan “Şoför” ve “Kullanıcı sınıfları” türetilmiştir. “Kullanıcı sınıfı”nın Firma niteliği “Firma sınıfı”nı işaret eder. “Adres sınıfı” kullanıcının niteliğidir. Aralarında bire-bir ilişki vardır.



Şekil 38: İrsaliye sınıf diyagramı.

İrsaliye sınıf diyagramını incelediğimizde; olgu sınıfı “Giriş İrsaliye sınıfı”dır. Giriş İrsaliye sınıfının niteliklerini incelediğimizde, Gönderici ve alıcı niteliği “Tedarikci” ve “Alıcı” sınıfını işaret eder.

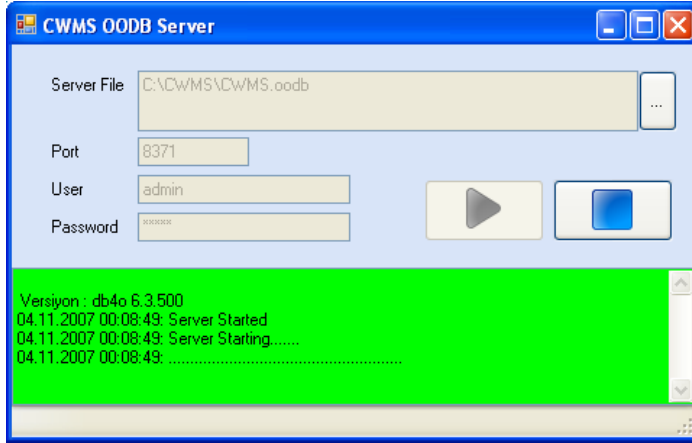
İrsaliyenin “Stok listesi” niteliği “Stoklar” sınıfını işaret eder. “Stoklar”, “Stok detay sınıfı”nın dizi halinde tutulmasından oluşmuş bir sınıftır. “Stok sınıfı” da “Stok detay”ın alt sınıfıdır.



Şekil 39 : Stok hareket sınıf diyagramı.

Stok hareket sınıf diyagramını incelendiğinde; “Stok hareket sınıfı”nın olgu tablosu olduğu açıkça görülmektedir. “Stok hareket sınıfı” “stok kayıtları”nı barındırır. “Tedarikçi”, “Alıcı” ve “Stok detay sınıfı” “Stok hareket sınıfı”nın alt sınıfıdır. Metotları incelendiğinde operasyona ilişkin metotlar da bulunmaktadır.

3.6. Örnek İşlemler

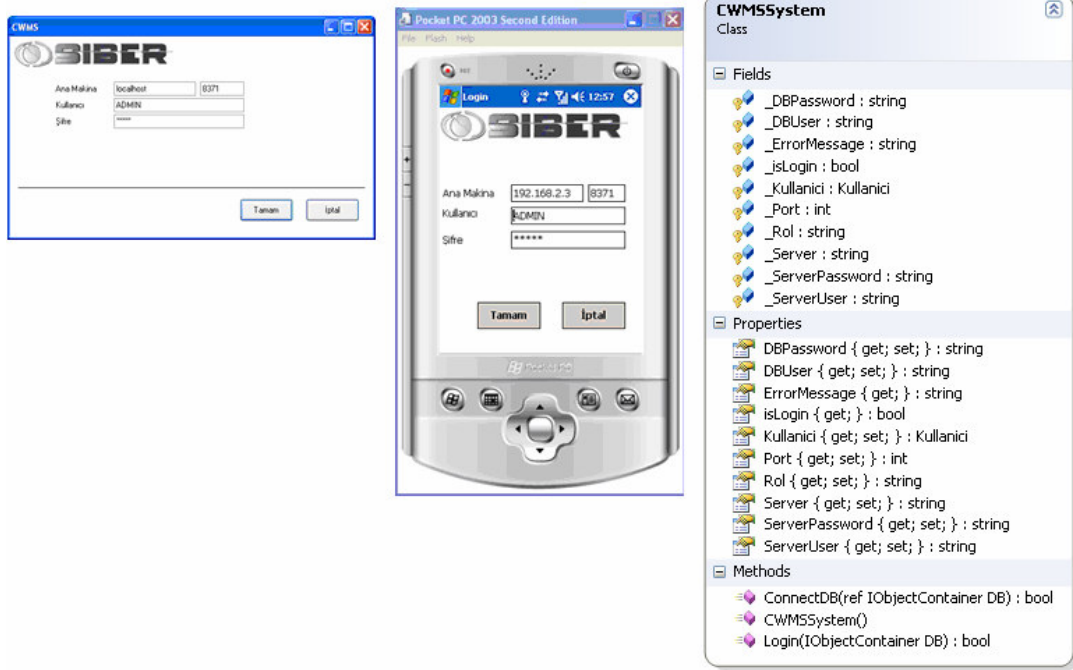


Şekil 40: Sunucu çalışması.

Sunucu çalıştırılır ve istemcilerin istekte bulunmasını bekler. Sunucu üzerinde sistemin çalışması için gerekli parametreler mevcuttur. *Server file*, fiziksel veri tabanını gösterir. *Port*, istemcilerin sunucuya bağlanacağı kanaldır.

```
try
{
    server = Db4oFactory.OpenServer(ConfigureCWMS(),EserverFile.Text,
        Int32.Parse(EPort.Text));

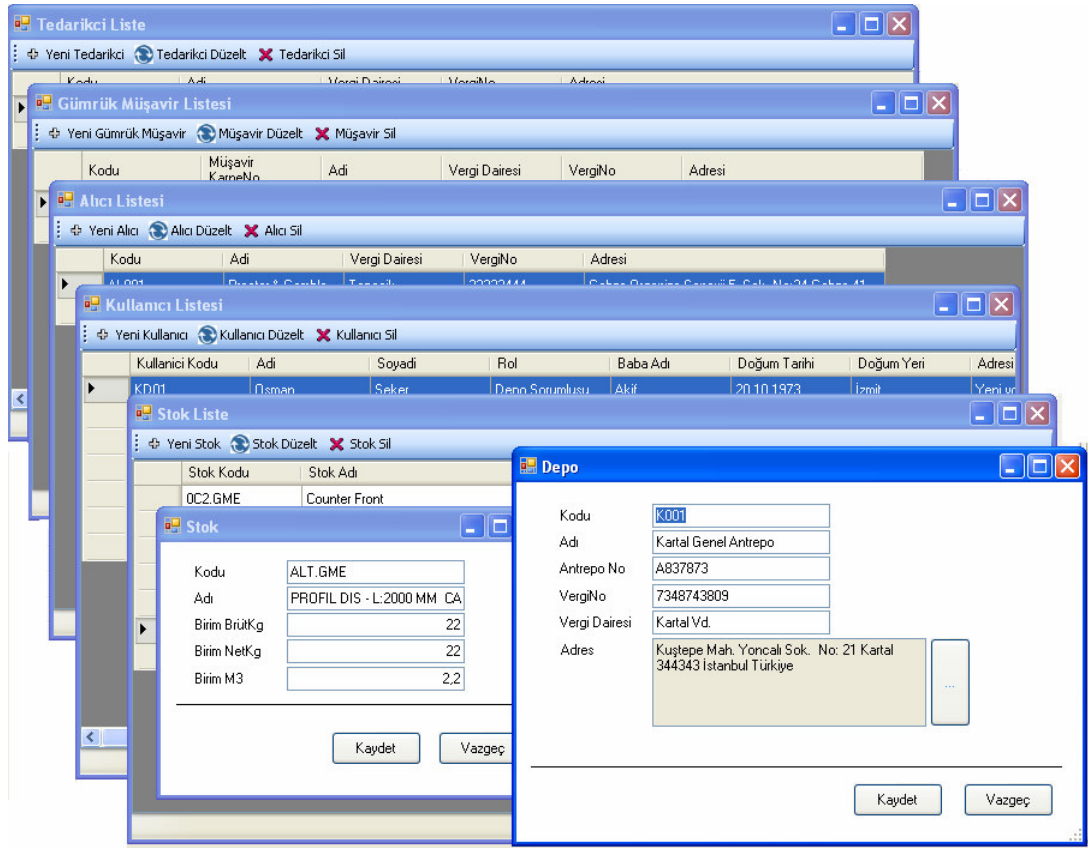
    server.GrantAccess(EUser.Text, EPassword.Text);
}
```



Şekil 41: İstemci giriş.

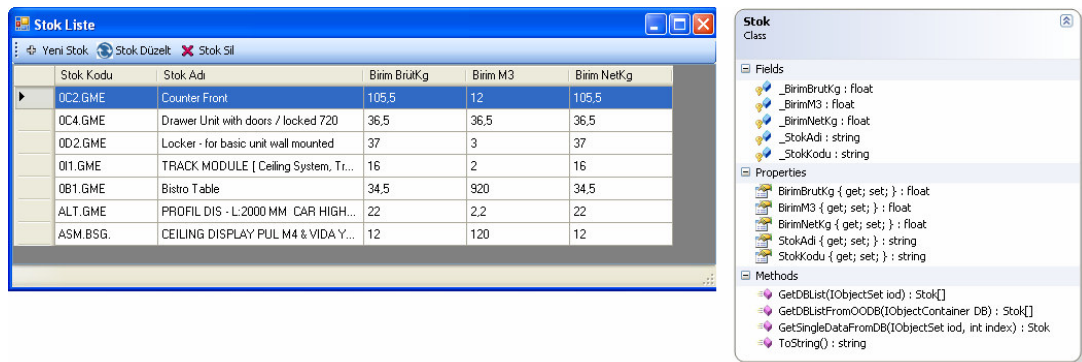
Mobil ve Masaüstü uygulaması olmak üzere iki farklı istemci mevcuttur. Her iki uygulama da istemciye bağlanma isteğinde bulunduğu anda, CWMSSystem sınıfından nesne yaratılır ve istemciye bağlanma ve yetki kontrol işlemi bu sınıfın ConnectDB ve Login metotları sayesinde gerçekleştirilir.

Şekil 41’de soldaki resim masaüstü uygulamanın giriş ekranıdır. Bu uygulama Depo Personeli, Tedarikci, Alıcı ve Gümrük Müşaviri tarafından kullanılacaktır. Mobil uygulama depo personeli tarafından stok yaratma, konum değiştirme ve sevkiyat için kullanılacaktır.



Şekil 42: Referans sınıfları.

Referans verileri sisteme girilerek uygulamanın çalışmasına başlanır. Veri kaydetme ve okuma işlemi aşağıdaki gibi yapılır:



Şekil 43: Stok kartları kaydetme, okuma, silme

Stok sınıfının GetDBListFromOODB metodu sunucudan stok nesnesinin tüm kayıtlarını ister ve bu kayıtları GetDBList metodunu kullanarak, şekil 43’de görülen listenin dizisine referans olarak vererek ekranda görünmesi sağlanır.

```
public static Stok[] GetDBListFromOODB(IObjectContainer DB)
{
    IQuery query = DB.Query();
    query.Constrain(typeof(Stok));
    IObjectSet result = query.Execute();
    return GetDBList(result);
}

public static Stok[] GetDBList(IObjectSet iod)
{
    Stok[] klist = new Stok[iod.Count];
    for (int i = 0; i < iod.Count; ++i)
    {
        klist[i] = (Stok)iod[i];
    }
    return klist;
}
```

“Stok Listesi” ekranında “Yeni Stok” düğmesine bastığımız zaman bu işlem kullanıcı ile nesne arasındaki iletişimi sağlayan “Stok Formu”nu ekrana getirerek kullanıcının veri girmesini sağlar.

```

try
{
    Stok stok = new Stok();
    FormStok fkl = new FormStok();
    fkl.stok = stok;
    fkl.ShowDialog();
    if (fkl.DialogResult == DialogResult.OK)
    {
        stok = fkl.stok;
        FormLogin.DB.Set(stok);
        FormLogin.DB.Commit();
        stokBindingSource.DataSource =
        Stok.GetDBListFromOODB(FormLogin.DB);
    }
}
catch (Exception ex)
{
    FormLogin.DB.Rollback();
    MessageBox.Show(ex.Message);
    FormStokListe_Load(sender, e);
}

```

Kullanıcı veriyi girip, “Kaydet” düğmesine bastığında verinin kaydetme isteği sunucuya bildirilir.

Kullanıcı, girdiği bir verinin değiştirilmesini istiyorsa, “Stok Düzeltme” butonuna basarak üzerinde bulunduğu nesnenin detayını görür ve değiştirir. Temel olarak yeni stok tanımlama ile aynıdır. Tek fark; yeni stok denildiğinde stok sınıfından yeni nesne yaratılması ve değer atama işleminin bu nesneye yapılmasıdır. Düzeltme işleminde var olan nesnenin nitelikleri değiştirilir.

Kayıt silme işleminde aşağıdaki kod işletilir. Listenin dizisinde bulunan nesne, tip dönüştürme işlemi yapılarak “Stok Sınıfı”na dönüştürülür ve sunucuya silme isteği gönderilir.

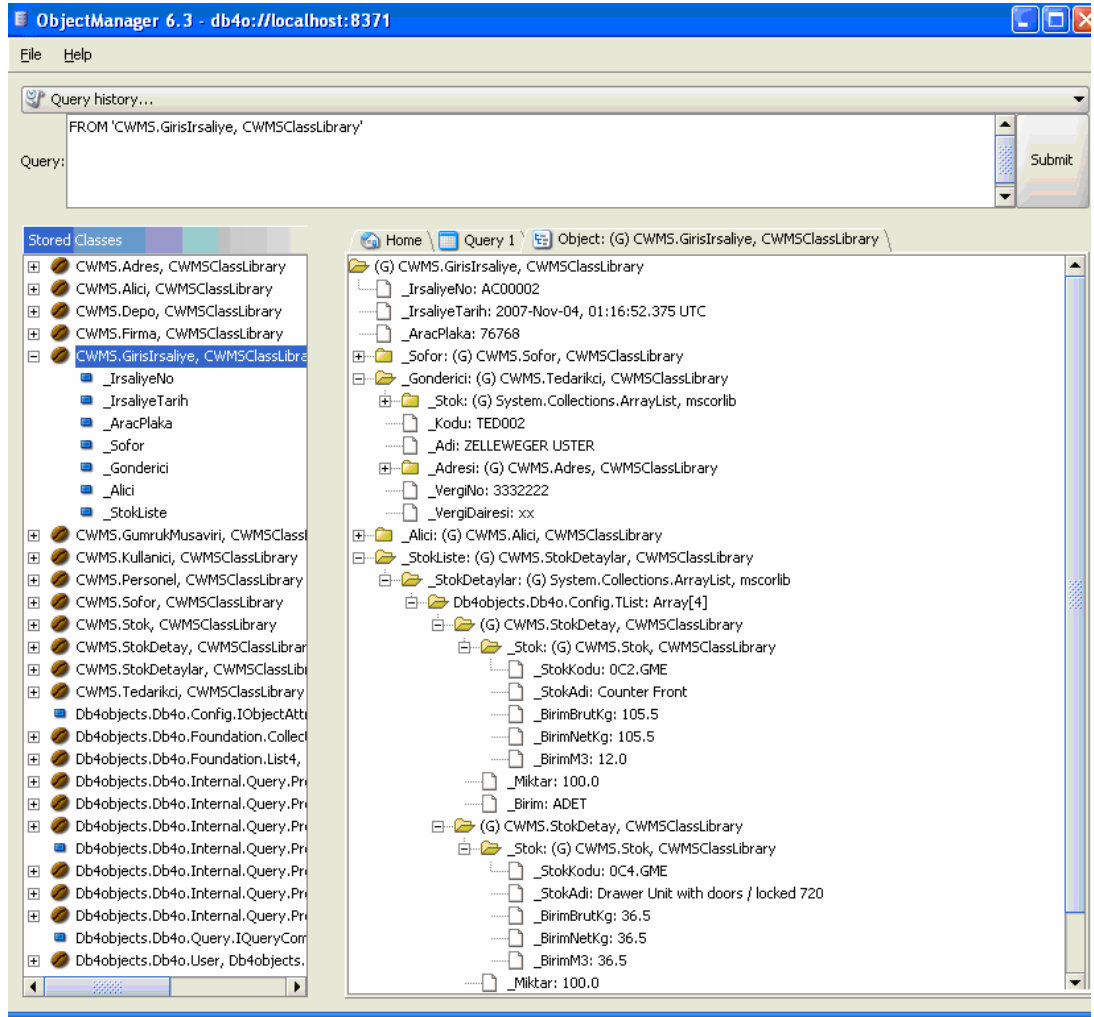
```
if (MessageBox.Show("Kayıt silinecek", "Sil", MessageBoxButtons.YesNo) =  
= DialogResult.Yes)  
{  
    Stok k = (Stok)dataGridView1.SelectedRows[0].DataBoundItem;  
    FormLogin.DB.Delete(k);  
    FormLogin.DB.Commit();  
    stokBindingSource.DataSource =  
    Stok.GetDBListFromOODB(FormLogin.DB);  
}
```

The screenshot shows the 'Giriş İrsaliyesi' (Entry Receipt) application interface. The window is titled 'Giriş İrsaliyesi' and contains a form for entering receipt details. The form includes fields for 'İrsaliye No' (Receipt No), 'İrsaliye Tarihi' (Receipt Date), 'Plaka' (Plate), 'Şoför' (Driver), 'Tedarikçi' (Supplier), and 'Alıcı' (Buyer). Below the form is a table with columns 'Stok' (Stock), 'Miktar' (Quantity), and 'Birim' (Unit). The table contains four rows of stock items. At the bottom of the window are buttons for 'Kaydet' (Save) and 'Vazgeç' (Cancel).

İrsaliye No	İrsaliye Tarihi	Plaka	Şoför	Alici	Gönderici
AB00001	04.11.2007 01:13	2347US768	Ahmet Arda	AL002 Dephi Packard	TED001 SIMEONIDIS ...
AC00002	04.11.2007 01:16	76768	Halil Önder	AL001 Procter & Gamble	TED002 ZELLEWEGER...

Stok	Miktar	Birim
OC2.GME Counter Front	100	ADET
OC4.GME Drawer Unit with doors / locked 720	100	ADET
OD2.GME Locker - for basic unit wall mounted	10	ADET
ALT.GME PROFIL DIS - L:2000 MM CAR HI...	10	ADET

Şekil 44: Giriş irsaliyesi kullanıcı arayüzü.



Şekil 45: Object Manager nesne görünümü

Şekil 45’de Giriş İrsaliye nesnelerinin kullanıcı arayüzü gözükmektedir.

Giriş İrsaliye nesnesi karmaşık bir nesnedir. Arayüz sayesinde kullanıcıya sade bir şekilde gösterilmiştir.

Şekil 45’de Object Manager yardımcı programı ile nesnenin karmaşık yapısı gösterilmiştir. Bu yardımcı program sayesinde nesnenin nitelikleri ve alt sınıfları detaylı şekilde gözükmektedir.



Şekil 46: Stok giriş.

Şekildeki “Stok Giriş Ekranı” mobil uygulamadır. Bu uygulamanın, “Pocket PC Emulator” programı üstünde çalışması sağlanmıştır.

Uygulamadaki yararı ise; araç rampaya boşaltma işlemi yaptığı zaman, stok girişlerinin fiziksel stoğun yanında yapılmasını sağlayarak hata riskini azaltmasıdır.

Operasyon - :

İşemri Durum: İthalat Bekleyen Stoklar

Tedarikçi: TED001 SIMEONIDIS PANAGIOTI

Alıcı: ... S

PaletNo: ... S

Stok Kodu: ...

İsemri No: ...

Ön FaturaNo: ...

FaturaNo: FT12333

Özet Beyan No: ...

AntrepoNo: ...

İthalatNo: ...

SevkiyatNo: ...

Bul

İşemri Oluştur İşemri İptal Et İthalat Gerçekleştir İthalat İptal

	PaletNo	GC	StokDetay	ÖnFaturaNo	FaturaNo	AntBeyanNo	ÖzetBeyanNo	İthalatİsemriNo	İthalatİse
	8	Giris	0I1.GME 3 ADET	FT0001	FT12333	07230330AN2099	3400300LK387883	IS0001	05.11.20
	5	Giris	0D2.GME 2 ADET	FT0001	FT12333	07230330AN2099	3400300LK387883	IS0001	05.11.20
	6	Giris	0I1.GME 3 ADET	FT0001	FT12333	07230330AN2099	3400300LK387883	IS0001	05.11.20
	7	Giris	0B1.GME 8 ADET	FT0001	FT12333	07230330AN2099	3400300LK387883	IS0001	05.11.20
	1	Giris	0C2.GME 1 ADET	FT0001	FT12333	07230330AN2099	3400300LK387883	IS0001	05.11.20
	2	Giris	0C4.GME 2 ADET	FT0001	FT12333	07230330AN2099	3400300LK387883	IS0001	05.11.20
	3	Giris	0D2.GME 2 ADET	FT0001	FT12333	07230330AN2099	3400300LK387883	IS0001	05.11.20
	4	Giris	0D2.GME 2 ADET	FT0001	FT12333	07230330AN2099	3400300LK387883	IS0001	05.11.20

Şekil 47: Sorgulama.

Şekil 47'deki sorgulama ekranı aynı zamanda operasyonel işlemlerin yapıldığı modüldür. Bu modül tüm birimler tarafından kullanılmaktadır.

Bu modülde sorgulama yapabileceğimiz tüm nitelikler belirtilmiş ve bu niteliklere göre aşağıdaki kod işletilerek sorgulama yapılması sağlanmıştır.

```

IQuery query = FormLogin.DB.Query();
query.Constrain(typeof(StokHareket));

if (ElsemriDurum.Text == "İşemri Verilmemiş Stoklar")
{
    query.Discard("_GC").Discard("Value__").Constrain(StokHareket.GirisCikis.Giris);
    query.Discard("_IthalatIsemriNo").Constrain(null);
}
else if (ElsemriDurum.Text == "Ithalat Bekleyen Stoklar")
{
    query.Discard("_GC").Discard("Value__").Constrain(StokHareket.GirisCikis.Giris);
    query.Discard("_IthalatIsemriNo").Constrain(null).Not();
    query.Discard("_IthalatNo").Constrain(null);
}
else if (ElsemriDurum.Text == "Sevk Bekleyen Stoklar")
{
    query.Discard("_GC").Discard("Value__").Constrain(StokHareket.GirisCikis.Giris);
    query.Discard("_IthalatIsemriNo").Constrain(null).Not();
    query.Discard("_IthalatNo").Constrain(null).Not();
    query.Discard("_SevkiyatNo").Constrain(null);
}
else if (ElsemriDurum.Text == "Sevk Edilen Stoklar")
{
    query.Discard("_GC").Discard("Value__").Constrain(StokHareket.GirisCikis.Cikis);
    query.Discard("_IthalatIsemriNo").Constrain(null).Not();
    query.Discard("_IthalatNo").Constrain(null).Not();
    query.Discard("_SevkiyatNo").Constrain(null).Not();
}
if (EFaturaNo.Text != "")
    query.Discard("_FaturaNo").Constrain(EFaturaNo.Text);

if (ElsemriNo.Text != "")
    query.Discard("_IthalatIsemriNo").Constrain(ElsemriNo.Text);

if (EOzetBeyanNo.Text != "")
    query.Discard("_OzetBeyanNo").Constrain(EOzetBeyanNo.Text);

if (EStokKodu.Text != "")
    query.Discard("_StokDetay").Discard("_Stok").Discard("_StokKodu")
        .Constrain(EStokKodu.Text);
if (tedarikci != null)
    query.Discard("_Tedarikci").Constrain(tedarikci);
IObjectSet result = query.Execute();
stokHareketBindingSource.DataSource = StokHareket.GetDBList(result);

```


Sonuç ve Genel Değerlendirme

Bu çalışmada “Nesne Yönelimli Modelleme Dili” kullanılarak kavramsal tasarım yapılmış ve bu tasarıma dayalı olarak bir nesneye yönelik veri tabanı yaratılmıştır. Bu çalışma sonunda programlama süresinin kısaldığı gözlemlenmiştir.

“Nesneye Yönelik Veri Tabanı” kullanıldığı için “Normalleştirme” ve “Bütünleştirme” işlemi yapılmasına gerek kalmamıştır.

Nesne yönelimli tasarımın problemi çözmede çok başarılı olduğu gözlemlenmiş ve programın parçalar şeklinde tasarlanması sağlanarak sistem karışıklığı giderilmiştir. Bu sayede programa genişlenebilirlik özelliği sağlanmıştır.

Karmaşık yapıya sahip nesneler üzerinde gözlem ve güncelleme işlemi yapılırken nesnenin yapısının karışıklığı kod yazım aşamasını zorlaştırmamıştır. Tasarım aşamasında hazırlanmış sınıflar üzerinde çalışarak gözlem ve güncelleme işlemi yapılmıştır.

Bu çalışmanın uygulama bölümünde “Genel Antrepo Yönetim Sistemi”nin başlıca sorunu olan bilgi paylaşım sorununun çözüldüğü, Genel Antrepolarla etkileşimde bulunan diğer birimlerin verimliliğinin arttığı gözlemlenmiştir. Klasik depo yönetimi ile ilgili prosedürlerin programa dahil edilmesi ile birlikte programın daha kullanışlı hale geleceği düşünülmüştür.

Kaynakça

Babür, Z., *Bilgisayarla İletişim*, Sistem Yayıncılık A.Ş., İstanbul, 1995

Shepherd, J. C., Database Management, Irwin, Homewood IL, 1990

Stephenes, R., Plew, R., Sams Teach Yourself Beginning Databases in 24 Hours, Sams Pub., 2003

Özkan, Y., Veri Tabanı Sistemleri, Alfa Ltd.Şti., İstanbul, 2003

Jefery, D. R., Lawrence, M. J., Systems Analysis and Design, Prentice-Hall of Australia Pty Ltd., 1984

Bruch, J., Grundnitski., Information Systems Theory and Practice, John Wiley & Sons Inc., Canada, 1989

Rolland, F. D., *The Essence Of Databases*, Prentice-Hall Europe, Hertfordshire, 1998

Yarımağan, Ü., Veri Tabanı Sistemleri, Akademi Yay.Ltd. Şti., 2000

Howe, D. R., *Data Analysis for Data Base Design*, Edward Arnold Pty Ltd., Australia, 1983

Bontempo, C., Saracco, J., Maro, C., *Database Management Principles and Products*, Prentice Hall Inc., Usa, 1995

Senn, J. A., Analysis & Design of Information Systems, MacGraw-Hill Book Co., Singapore, 1989

Hawryszkiewicz, I., Introduction to Systems Analysis and Design, Prentice-Hall of Australia Pty. Ltd., 1998

Hurson, A.R., Parallel Architectures for Database Systems, IEEE Computer Society Press Washington D.C., 1989

<http://www.dis.port.ac.uk/~chandler/OOlectures/database/database.htm>, 2002

<http://www.soc.napier.ac.uk/module.php3?op=getresource&cloaking=no&resourceid=746239> 0, Object-Oriented Databases, Module No: CO42009, Group Seminar:OODB System Comparison

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., Object-Oriented Modeling And Design, Prentice-Hall, New Jersey, 1991

Bekke, J. H., *Semantic Data Modeling*, Prentice Hall International Ltd., Hertfordshire, 1992

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen W.

<http://www.dis.port.ac.uk/~chandler/OOlectures/database/database.htm>, 2002

http://www.pjwstk.edu.pl/~habela/publications/ADVIS2006Three-level_OODBA.pdf (Advanced in Information Systems, 4th International Conference, ADVIS2006, Izmir, Turkey, October, 18-20 2006)

<http://www.dis.port.ac.uk/~chandler/OOlectures/database/database.htm>, 2002

Zengin, A., Visual C#2005, Nirvana, Ankara, 2006

ÖZGEÇMİŞ

1. Bireysel Bilgiler

Adı : Hasan
Soyadı : ERDEMİR
Doğum Yeri / Tarihi : İzmit 1976
Uyruğu : T.C.
Adres / Telefon : Yukarı Dudullu Mah. Çetin Cad. Garip Sok. Deniz Apt.No:21
D:2 Ümraniye İSTANBUL 0533 6083192

2. Eğitim

Beykent Üniversitesi
Fen Bilimleri Enstitüsü Bilgisayar Ağları ve İnternet Teknolojileri (2004–2008)

Anadolu Üniversitesi
İşletme Fakültesi İşletme Bölümü (1999–2003)

Muğla Üniversitesi
Muğla Meslek Yüksek Okulu Bilgisayar Programcılığı (1994–1998)

Körfez Endüstri Meslek Lisesi
Elektronik Bölümü (1989–1993)

3. Yabancı Dil

İngilizce; İyi

2. Ünvanlar

B.Sc, Anadolu Üniversitesi İşletme Fakültesi İşletme Bölümü 2003

5. Mesleki Deneyim

Barsan Global Lojistik
Yazılım Uzmanı – Proje Yöneticisi (2002-2007)
Yön Bilgi Sistemleri
Yazılım Uzmanı (1998-2001)