

# DERS NOTLARI

**Ders Kodu** : 001 (END314)

**Ders Adı** : Sistem Analizi ve Tasarımı

## **Kısa Ders Özeti**

Bu ders, genel sistem yaklaşımı kavramlarını ve işletmelerin sistem yaklaşımı ile incelenmesini gösterdikten sonra, bilgisayara dayalı bilgi sistemlerini anlatır. Bilgisayara dayalı bilgi sistemlerinin tasarlanması için gerekli araçlar dersin kapsamında bulunmaktadır.

## **Dersin Hedefleri**

Bu ders sonunda öğrenciler, işletmeleri sistem yaklaşımı ile incelemeyi ve bu yaklaşımla problemlere çözüm getirmeyi öğreneceklerdir. Ayrıca öğrenciler, bir yönetim bilgi sisteminin nasıl tasarlanacağını da görecekle ve yapacakları bir proje yardımıyla bu tasarımla ilgili becerileri de kazanacaklardır.

## **Kullanılan Kaynak Kitaplar**

Sistem Analizi (Doç. Dr. Haluk Erku - Kıy Yayınları 1989)

İşletme Yönetiminde Sistem Yaklaşımı (Prof. Dr. H. Öner Esen – Alfa Basım Yayın Dağıtım 1998)

Yönetim Bilgi Sistemleri (Doç. Dr. Hadi Gökçen – EPI Yayıncılık 2002)

# Ders İçerik Planı (Layout)

<b>1. SİSTEM KAVRAMI</b> .....	<b>1</b>
1.1. SİSTEM DÜŞÜNCESİNİN ORTAYA ÇIKIŞI.....	1
1.2. SİSTEM TANIMI VE BİLEŞENLERİ .....	3
1.2.1 Ögeler.....	4
1.2.2 İlişkiler .....	4
1.2.3 Amaçlar.....	5
1.3. GENEL BİR SİSTEMİN ŞEMATİK GÖSTERİMİ .....	5
1.4. SİSTEM HİYERARŞİSİ .....	6
1.5. SİSTEM SINIFLANDIRMASI .....	6
1.5.1 Açık ve kapalı sistemler.....	7
1.5.2 Canlı ve cansız sistemler.....	7
1.5.3 Doğal ve insan yapısı sistemler.....	7
1.5.4 Statik ve dinamik sistemler.....	7
1.5.5 Soyut ve somut sistemler .....	7
1.5.6 Basit ve karmaşık sistemler .....	7
1.6. ENDÜSTRİ MÜHENDİSLİĞİ VE SİSTEM YAKLAŞIMI.....	8
<b>2. SİSTEM MODELLERİ</b> .....	<b>9</b>
2.1. SÖZLÜ (KAVRAMSAL) MODELLER.....	9
2.2. ŞEMATİK MODELLER .....	9
2.2.1 Grafikler.....	9
2.2.2 Gantt Şeması .....	10
2.2.3 Ağ Diyagramı.....	10
2.2.4 Karar Ağacı.....	11
2.2.5 Organizasyon Şeması.....	12
2.2.6 Süreç Akış Şeması .....	12
<b>3. SİSTEM ANALİZİ</b> .....	<b>15</b>
3.1. SİSTEM ANALİZİ AŞAMALARI .....	15
3.2. SİSTEM ANALİZİNİN TEMEL FAALİYETLERİ (GANTT ŞEMASI) .....	16
3.3. YENİ SİSTEME GEÇİŞ YAKLAŞIMLARI.....	16
3.3.1 Doğrudan Geçiş .....	17
3.3.2 Paralel Geçiş.....	17
3.3.3 Safhali (adım adım) Geçiş.....	17
3.3.4 Pilot Geçiş.....	17
<b>4. BİLGİ SİSTEMLERİNE GİRİŞ (YBS)</b> .....	<b>18</b>
4.1. VERİ VE BİLGİ.....	18
4.2. BİLGİNİN KARAKTERİSTİKLERİ.....	18
4.3. YÖNETİM VE KARAR VERME SEVİYELERİ.....	19
4.4. YÖNETİMDE BİLGİ KULLANIMI.....	20
4.5. KARAR TIPLERİ.....	20
4.6. KARAR VERME PROSESİ .....	21
4.7. BİLGİ SİSTEMLERİ.....	22
4.7.1 Bilgi sistemlerinin faydaları.....	23
<b>5. BİLGİSAYARA DAYALI BİLGİ SİSTEMLERİ</b> .....	<b>24</b>
5.1. KAYIT/VERİ İŞLEME SİSTEMLERİ (VİS).....	24
5.2. YÖNETİM BİLGİ SİSTEMLERİ (YBS).....	25
5.2.1 YBS ve VİS'in farklılıkları .....	26
5.2.2 YBS ve iş fonksiyonları.....	27
5.2.3 YBS ve Veritabanı Yönetim Sistemi (VTYS).....	27
5.3. KARAR DESTEK SİSTEMLERİ (KDS) .....	28
5.4. OFİS OTOMASYON SİSTEMLERİ.....	28
5.5. YAPAY ZEKA VE UZMAN SİSTEMLER .....	28
5.5.1 Bilgiye dayalı ve uzman sistemler .....	29
5.5.2 Doğal Diller .....	29

5.5.3	İnsanın duyumsal yeteneklerinin taklidi.....	29
5.5.4	Robotik.....	29
5.6.	ÜST YÖNETİM DESTEK SİSTEMLERİ.....	29
<b>6.</b>	<b>BİLGİ SİSTEMLERİNİN GELİŞTİRİLMESİNDE KULLANILAN ARAÇLAR.....</b>	<b>31</b>
6.1.	AKIŞ ŞEMALARI.....	31
6.2.	VERİ AKIŞ DİYAGRAMLARI (VAD).....	31
6.2.1	VAD Çizim Kuralları.....	33
6.2.2	Mantıksal-Fiziksel VAD.....	36
6.3.	YAPISAL ŞEMALAR.....	37
6.4.	YAPISAL DİL.....	37
6.5.	KARAR TABLOLARI.....	40
6.6.	KARAR AĞAÇLARI.....	41
6.7.	HIPO.....	42
<b>7.</b>	<b>VERİTABANI TASARIMI VE YÖNETİMİ.....</b>	<b>44</b>
7.1.	VERİ YÖNETİMİ.....	44
7.1.1	Veri Hiyerarşisi.....	44
7.1.2	Birimler (Entity), Özellikler (Attributes) ve Anahtarlar.....	44
7.1.3	Veri yönetimi yaklaşımları.....	45
7.1.3.1	Geleneksel yaklaşım.....	45
7.1.3.2	Veritabanı (VT) yaklaşımı.....	46
7.2.	VERİTABANI MODELLERİ.....	47
7.2.1	Hiyerarşik (Ağaç) Yapılar.....	47
7.2.2	Ağ Yapılar.....	47
7.2.3	İlişkisel Yapılar.....	48
7.3.	VTYS BİLEŞENLERİ.....	49
7.3.1	Veri Sözlüğü (VS) Kullanma.....	50
7.3.1.1	Veri akışı sözlük girişi.....	50
7.3.1.2	Veri Deposu Sözlük Girişi.....	51
7.3.1.3	Proses Sözlük Girişi.....	51
7.3.1.4	Veri Yapısı Sözlük Girişi.....	51
7.3.1.5	Veri Elemanı Sözlük Girişi.....	52
7.4.	İLİŞKİSEL VERİTABANI TASARIMI.....	54
7.4.1	Birim İlişki Diyagramları.....	54
7.4.2	Normalizasyon için Yedi Adım Metodu.....	57
<b>EK1</b>	<b>– ÇEŞİTLİ VAD ÖRNEKLERİ.....</b>	<b>58</b>
<b>EK2</b>	<b>– ÇEŞİTLİ YAPISAL ŞEMA ÖRNEKLERİ.....</b>	<b>65</b>
<b>EK3</b>	<b>– NORMALİZASYON İÇİN YEDİ ADIM METODU.....</b>	<b>70</b>

# ŞEKİL LİSTESİ

Şekil 1.1 – Bilimsel Evrim .....	3
Şekil 1.2 – Bir sistemin en temel hali.....	5
Şekil 1.3 – Geri beslemeli sistem (dinamik sistem).....	5
Şekil 1.4 – Detaylı Sistem Gösterimi .....	6
Şekil 1.5 – İşletme Açısından Sistem Hiyerarşisi .....	6
Şekil 2.1 – Bir Çubuk Grafik Örneği .....	9
Şekil 2.2 – Başabaş Noktası Grafiği.....	10
Şekil 2.3 – Gantt Şeması Örneği.....	10
Şekil 2.4 – Bir Ağ Diyagramı Örneği.....	11
Şekil 2.5 – Bir Karar Ağacı Örneği .....	11
Şekil 2.6 – Bir Organizasyon Şeması Örneği.....	12
Şekil 2.7 – Bir Süreç Akış Şeması Örneği (Telefonla Satış) .....	13
Şekil 3.1 – Sistem Analizi Aşamaları.....	15
Şekil 3.2 – Sistem Analizi Faaliyetleri.....	16
Şekil 3.3 – Geçiş Yaklaşımları.....	16
Şekil 4.1 – Karar Verme (Yönetim) Seviyeleri.....	19
Şekil 4.2 – Yönetim Seviyelerinin Bilgi İhtiyaçları.....	20
Şekil 4.3 – Yönetim Seviyelerinde Karar Tipleri .....	21
Şekil 4.4 – Karar Verme Prosesi .....	21
Şekil 4.5 – Yöneticinin Karar Verme Prosesi.....	22
Şekil 4.6 – Bilgi Sistemi Öğeleri ve İlişkileri.....	22
Şekil 5.1 – Bir VİS Uygulamasının Yapısı .....	24
Şekil 5.2 – VİS, YBS ilişkisi .....	26
Şekil 5.3 – YBS ve İş Fonksiyonları .....	27
Şekil 5.4 – YBS ile VTYS arasındaki ilişki .....	27
Şekil 5.5 – ÜDS'nin diğer sistemlerle ilişkisi.....	30
Şekil 6.1 – VAD Sembolleri .....	31
Şekil 6.2 – Sistem Akış ve Program Akış Şeması Örnekleri.....	32
Şekil 6.3 – Genel Bir VAD .....	33
Şekil 6.4 – VAD'da Seviyelendirme.....	34
Şekil 6.5 - Çizimindeki Yanlışlıklar ve Doğru Çizim Yolları .....	36
Şekil 6.6 – Dönüştürme ve İşleme Merkezli Sistemler .....	37
Şekil 6.7 – Karar ağacı örneği.....	42
Şekil 6.8 – Bordro Sistemi için HIPO Hiyerarşi Diyagramı .....	43
Şekil 6.9 – HIPO Özet Diyagramı.....	43
Şekil 7.1 – Veri Hiyerarşisi.....	44
Şekil 7.2 – Geleneksel Veri Yönetimi Yaklaşımı .....	45
Şekil 7.3 – Veritabanı Yaklaşımı .....	47
Şekil 7.4 – VTYS'nin Bileşenleri .....	49
Şekil 7.5 – Veri Akış Sözlük Girişi Örnekleri.....	50
Şekil 7.6 – Veri Deposu Sözlük Girişi.....	51
Şekil 7.7 – Proses Sözlük Girişi .....	51
Şekil 7.8 – Veri Yapısı Sözlük Girişi .....	52
Şekil 7.9 – Veri Elemanı Sözlük Girişi .....	53
Şekil 7.10 – Veri Elemanı Sözlük Girişi ve Kod Tablosu .....	54
Şekil 7.11 – Birimler ve İlişkiler.....	55
Şekil 7.12 – BİD'in Farklı Gösterimleri.....	56
Şekil 7.13 – Örnek BİD .....	56
Şekil 7.14 – Üst ve Alt Tipler .....	57

## TABLO LİSTESİ

Tablo 1.1 – Bazı sistem örnekleri .....	4
Tablo 2.1 – Süreç Akış Şeması Sembolleri (NCC) .....	12
Tablo 6.1 – İndirim Hesaplama Prosesi İçin Karar Tablosu .....	40
Tablo 6.2 – İndirgenmiş Karar Tablosu .....	41
Tablo 6.3 – Şartların ikiden fazla olasılık taşıdığı durum için örnek (Karar Tablosu).....	41
Tablo 7.1 – Öğrenci Dosyası (Veri Birimi).....	45
Tablo 7.2 – Veri Tekrarlılığı ve Dosyalar Arası Bağlantı Kopukluğu .....	46
Tablo 7.3 – Geleneksel Kayıt Yerleşimli bir Öğrenci Kayıt Dosyası .....	48
Tablo 7.4 – İlişkisel Veri Tabanı .....	48
Tablo 7.5 – Bir Sorgulama Sonuç Raporu.....	49

# DERS NOTLARI

## 1. SİSTEM KAVRAMI

Çeşitli sistem tanımları:

- Birden çok şey veya parçaların kombinasyonu veya bir araya getirilmesi sonucunda oluşan karmaşık veya bölünmez bütündür.
- Birbirleri ile etkileşimli elemanların oluşturduğu topluluktur.
- Nesnelere ve bu nesnelere ile özelliklerinin arasındaki ilişkilerin meydana getirdiği topluluktur.
- Aralarında ilişkiler olan parçaların oluşturduğu topluluktur.
- Plana uygun bir amacı gerçekleştirmek üzere tasarlanmış çeşitli bileşenlerin oluşturduğu bütündür.
- Bir işletmede bir faaliyeti gerçekleştirmek amacıyla bütünleştirilmiş bir plan oluşturmak üzere birbirleri ile ilişkili çeşitli süreçlerin oluşturduğu bir şebekedir.
- Birbirleri ile ilişkili bileşenlerin oluşturduğu karmaşık bütündür.

Bu tanımları çoğaltmak mümkündür. Fakat sonuçta bu ve benzeri tanımların ortak noktalarından faydalanılarak şu şekilde genel bir sistem tanımı yapmak uygun olacaktır:

**Sistem;** bir veya daha fazla amaca yada sonuca ulaşmak üzere bir arada bulunan ve aralarında ilişkiler olan fiziksel ya da kavramsal birden çok bileşenin (öğenin) oluşturduğu bütündür.

### 1.1. Sistem Düşüncesinin Ortaya Çıkışı

Sistem düşüncesindeki temel gelişmeler ve olayların sistem görüşü ile incelenmesi isteklerinin ortaya çıkışı 1940'lı yıllara rastlamaktadır. Bilim tarihine bakıldığında zaman en başta tüm bilimlerin felsefe içinde açıklandığı görülmektedir. Zaman içinde, arayan ve soran insan aklının sınırları belli inceleme alanlarına yönelip bu alanlara uygun araştırma yöntemleri geliştirilerek bilgi üretme gücünü elde etmesi sonucunda bilim felsefeden bağımsızlaşmıştır. Ardından bilim yarar üretme yönünde ilerleyerek teknoloji denen kavramı meydana çıkardı. Teknolojideki hızlı gelişmeler ve uzmanlaşma otomasyon kavramını ortaya çıkardı. Uzmanlaşma ve otomasyon, bir yandan verimlilik açısından iyileşme taleplerini karşılarken bir yandan da sorunların, sistemlerin ve işlevlerin giderek daha küçük parçalara ayrılmasına sebep oldu. Bu ayrılma ise birbirinden oldukça farklı sistemlerin ortak yanlarını, ortak işlevlerini, temel ilke ve niteliklerini anlamayı güçleştirdi. Bununla birlikte, sorunların birbirlerinden soyutlanmaları ve sanki birbirlerinden ilgisiz ve bağımsızmış gibi bir anlayışın ve yaklaşım alışkanlığının doğması sonucunu getirdi. Bu anlayış ve yaklaşım alışkanlığına karşı tepki olarak, sistemler arasındaki ortak ilkeleri, sorunları ve kavramları bilmek ve koordine etmek, insan makine sistemlerindeki büyüme ve karmaşıklaşmanın getirdiği sorunları aşmak amacıyla yeni bir yaklaşım ortaya çıktı. Bu yaklaşım sistem yaklaşımıdır.

Sistem düşüncesinin, diğer bir deyişle sistem yaklaşımının ortaya çıkmasına neden olan etmenler aşağıdaki gibi toparlanabilir:

1. **Bilimin bir bütün oluşu:** Bilim normalde bir bütündür. Bilimi ayrı disiplinler içinde incelemek onu daha iyi anlayabilmek için yapılmışsa da zaman içinde bütünlük bozulmuştur. Bilimi ayrı disiplinler içinde ele alıp incelemek disiplinlerin çevredeki olayları anlayabilmek açısından kısıtlı görüş açısı yüzünden yetersiz kalması sonucunu doğurmuştur. Sistem düşüncesi bu bütünlüğü disiplinler arası bir yaklaşımla aşmayı amaçlar.
2. **Bilimde savurganlık:** Yürütülen bilimsel çabalar kaynak savurganlığına yol açmaya başlamıştır. Farklı disiplinleri bir arada ilgilendiren konular her disiplin içinde ayrı ayrı ele alınıp incelenmekte ve bu yüzden gayretler gereksiz yere dağıtılmaktadır. Çoğu kez aynı sonuçlara ulaşılmakta bu da kaynak savurganlığını doğurmaktadır (jeofizik, fizikokimya, sosyo-ekonomi gibi). Sistem düşüncesi ile bu savurganlığın aşılması amaçlanmıştır.
3. **Bilimsel yöntemin yetersizliği:** Analiz ve senteze dayanan bilimsel yöntem, bilimin o gün itibarıyla ulaştığı noktada bilimsel problemlerin çözümü için yetersiz kalmıştır. Sistem düşüncesi içinde bilimsel yöntemi de içeren yeni bir yaklaşım önermiştir.
4. **Tükenmeyen sorunlar:** Yirminci yüzyılın ikinci yarısına gelindiğinde insanoğlunun sahip olduğu bilgiler çevredeki olayları çok küçük ayrıntılarına kadar çözmeye yetecek seviyede olmasına rağmen sorunlar ve sorunlar bitmemektedir. Sistem düşüncesi ile sorunların daha etkin ve hızlı çözülmesi amaçlanmıştır.

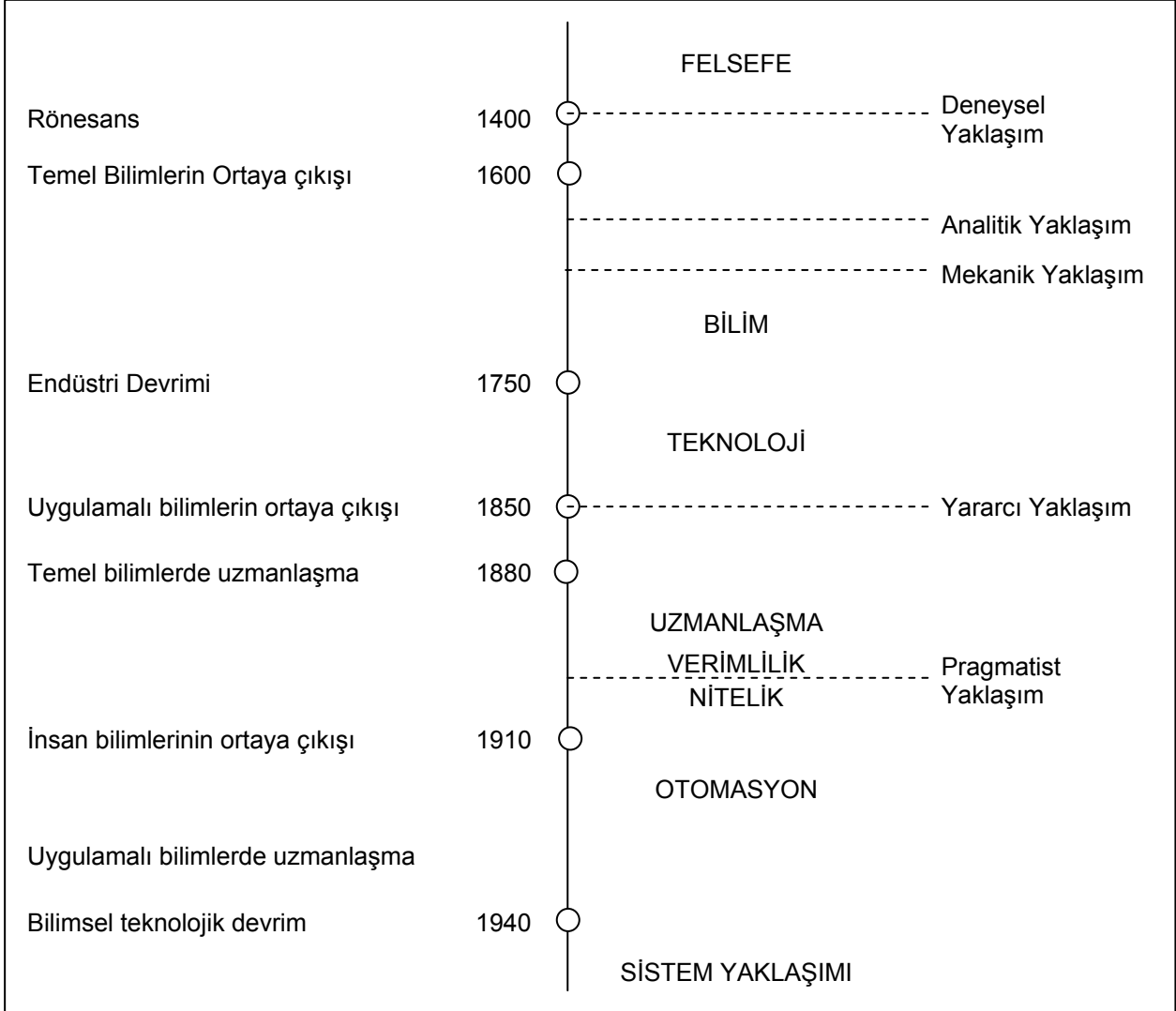
Yukarıda sayılan sebepler sonucu geliştirilen sistem yaklaşımının üç temel ilkesi vardır:

1. **Bütünsel Yaklaşım:** İncelenen sistem bir bütün olarak görülmelidir. Sistemin içerdiği sorunların birbirlerinden soyutlanamaz. Sistemin içerdiği bir öge ancak sistemin diğer öğeleriyle birlikte düşünüldüğünde işlevsel bir anlam ifade eder. Sistem, birbirleriyle etkileşimli öğelerden oluşmuş, çevresiyle etkileşimli bir bütünlüktür.
2. **Disiplinler Arası Yaklaşım:** Bütünsel yaklaşımın tamamlayıcısıdır, şöyle ki; incelenen sistemi bir bütün olarak görmenin ön koşulu ve aynı zamanda gerekli sonucu, o sisteme farklı görüş açılarıyla yaklaşabilmektir. Bu ön koşulu disiplinler arası yaklaşım sağlar. Eğer sorunlar üzerine tek bir bilim dalının görüş açısı ile gidilirse ön yargılı ve gerçek dışı sonuçlara varılması muhtemeldir. Disiplinleri insanlar ortaya çıkarmıştır ve disiplinler doğadaki sorulara farklı görüş açıları ile çözümler üretmeyi hedefler. Disiplinler arası yaklaşım sayesinde grup çalışması denen yöntem gündeme gelmiştir. Değişik bilim dallarında eğitim görmüş bilim adamları bir araya gelerek karar ve çözüm üretmeye yönelmişlerdir.
3. **Bilimsel Yaklaşım:** Sistem yaklaşımında sorunları bir bütün olarak görmenin ve sorunlara değişik görüş açılarıyla yaklaşmanın somut yöntemidir. Sistemler üzerinde çalışırken sorunların çözümü için bilimsel yöntem tercih edilir. Bu yöntem temel bilimler ve toplumsal bilimler açısından farklılık göstermektedir. Sistem analizinde sistemin işlevine göre bu yöntemlerden birisi kullanılabilir.

Temel bilimler için uygulanan bilimsel yöntem aşamaları şunlardır:

1. Olayın gözlenmesi, problemin tanımlanması
2. Hipotezin geliştirilmesi.

3. Veri ve bilgilerin toplanması
4. Deneyler yoluyla hipotezin test edilmesi
5. Hipotez hakkında sonuçlara varılması
6. Genelleme yardımıyla olayın kontrol altına alınması



Şekil 1.1 – Bilimsel Evrim

## 1.2. Sistem Tanımı ve Bileşenleri

Sistem, günümüzde çok sık kullanılan sözcüklerden birisidir. Hemen her türlü metinde bu sözcükle karşılaşmak olasıdır. Çevremizde olup biten her türlü faaliyet bir sistem olarak düşünülebilir. Böyle geniş anlamlar içeren bir sözcüğü tek bir tanımın içine sığdırmak güçtür. Gene de sistem olarak adlandırılan tüm kavramların içerdiği ortak noktalar bulunmaktadır. Bu noktalar öge, özellik, faaliyet ve durumdur. Bunları kısaca açıklarsak:

Öge : Sistem içindeki herhangi bir nesne

Özellik : Sistem içindeki ögelerin nitelikleri

Faaliyet : Sistemde değişimi sağlayan süreçler (prosesler)

**Durum** : Belli bir zaman noktasına sistemin öge, nitelik ve faaliyetlerinin tanımı.

Tablo 1.1’de bu kavramlarla ilgili örnekler verilmiştir.

<b>Sistem</b>	<b>Öğeler</b>	<b>Özellikler</b>	<b>Faaliyetler</b>
İmalat	Makine İşgücü Mamul	Hassas Nitelikli Bozuk	İmalat
Ulaşım	Taşıtlar Yol Levhalar	Hızlı Uzun Beyaz	Taşıma
İletişim	Mesajlar Cihazlar	Kısa Yeni	Haber gönderme

Tablo 1.1 – Bazı sistem örnekleri

Bu noktada sistem tanımına geri dönersek:

“Sistem; bir veya daha fazla amaca yada sonuca ulaşmak üzere bir arada bulunan ve aralarında ilişkiler olan fiziksel ya da kavramsal birden çok bileşenin (ögenin) oluşturduğu bütündür.” Bu tanıma göre:

- Sistem öğelerden oluşmuştur.
- Öğeler arasında ilişkiler vardır.
- Sistem belli bir amaca yönelmiştir.

Bu üç noktayı daha detaylı inceleyelim.

### 1.2.1 Öğeler

Sistemi meydana getiren fiziksel ya da kavramsal tüm bileşenler sistemin öğeleridir. Sistem içerisindeki faaliyetler de bir öge olarak düşünülebilir. Örneğin otomobili meydana getiren piston, direksiyon, vites vb. parçalar öğelerdir. Bir işletmedeki yönetim, denetleme gibi faaliyetlerde ögedir.

**Alt Sistem:** Sistemin içindeki onun yöneldiği amaca katkı sağlayan diğer sistemler alt sistemlerdir. Alt sistemler de sistemin öğeleridir.

### 1.2.2 İlişkiler

Sistem içerisindeki öğelerin birileri arasındaki her türlü akış ilişki olarak adlandırılır. Sistem öğeleri arasındaki ilişkiler değişik türden olabilirler. Bunlar:

#### Mekansal İlişki

Fiziksel öğelerin, bir mekan içinde belli bir ilişki içinde buldukları sistemler buna iyi bir örnektir. Örneğin bir imalat sistemindeki tezgahlar arasındaki uzaklık ilişkisi bir mekansal ilişkidir.

#### Zamansal İlişki

Sistem içinde olayların sırasını ayırt etmeye yarar. Örneğin bir arabanın hızı ile kat ettiği uzunluğu arasında bir zaman ilişkisi vardır. Ya da bir imalat sisteminde bir

mamulün izleyeceği işlem sıraları arasında bir zaman ilişkisi olabilir. Mamul X tezgahında işlenmeden Y tezgahında işlenemiyor olabilir.

### Neden Sonuç İlişkisi

Neden sonuç ilişkisi sistem öğeleri arasında da bulunan önemli bir doğa ilkesidir. Örneğin bir ekonomik sistemde bir ürünün fiyatı belirlenirken herhangi bir nedenle o ürüne olan talep artarsa bu nedenin sonucu olarak o ürünün fiyatı artacaktır.

### Enerjinin Korunumu İlişkisi

Maddenin bir biçimden diğerine geçerken enerji ve madde korunur. Bu ilişki bir doğa yasası olarak sistemler için de geçerlidir.

### Mantıksal İlişki

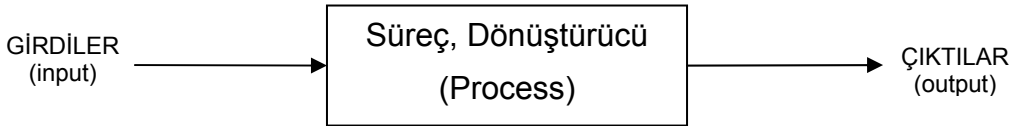
Özellikle soyut sistemlerde görülen bir ilişki türüdür ve bilgisayar programlarında sıkça rastlanır.

## 1.2.3 Amaçlar

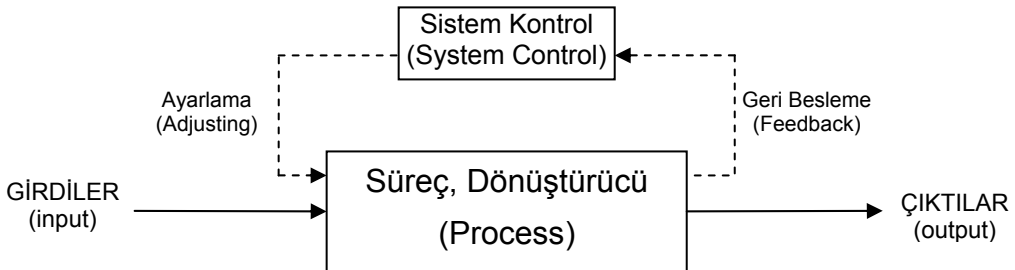
Her sistemin yöneldiği bir ya da daha fazla amaç vardır. Örneğin bir otomobil sistemi taşıma yapma amacına hizmet eder, üretim hattı imalat gerçekleştirir ya da bir eğitim sistemi insanları eğitmeyi amaçlar.

İnsan yapısı sistemler için amaçları tespit etmek çok zor değildir. Zaten bu sistemler bir amaca ulaşmak için insanlar tarafından üretilmiştir. Fakat, insan yapısı olmayan sistemler için amaçları tespit etmek her zaman kolay olmayabilir. İnsanın sindirim sisteminin amacının besinleri sindirip insana enerji sağlamak olduğunu söylemek kolaydır, ancak güneş sistemi gibi daha geniş sistemler için bu amacı tespit edebilmek teolojik tartışmalara neden olmaktadır.

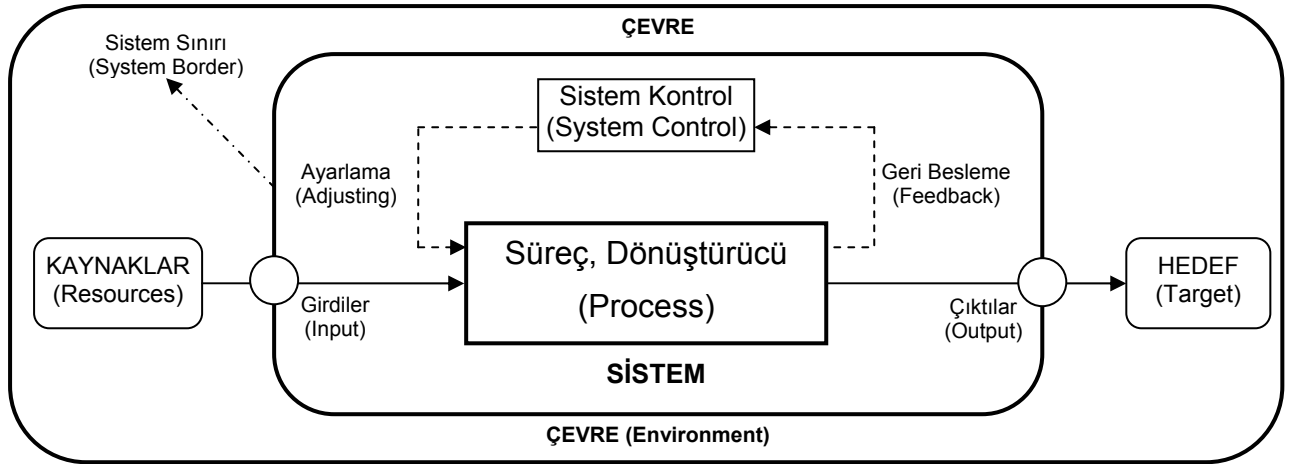
## 1.3. Genel bir Sistemin Şematik Gösterimi



Şekil 1.2 – Bir sistemin en temel hali



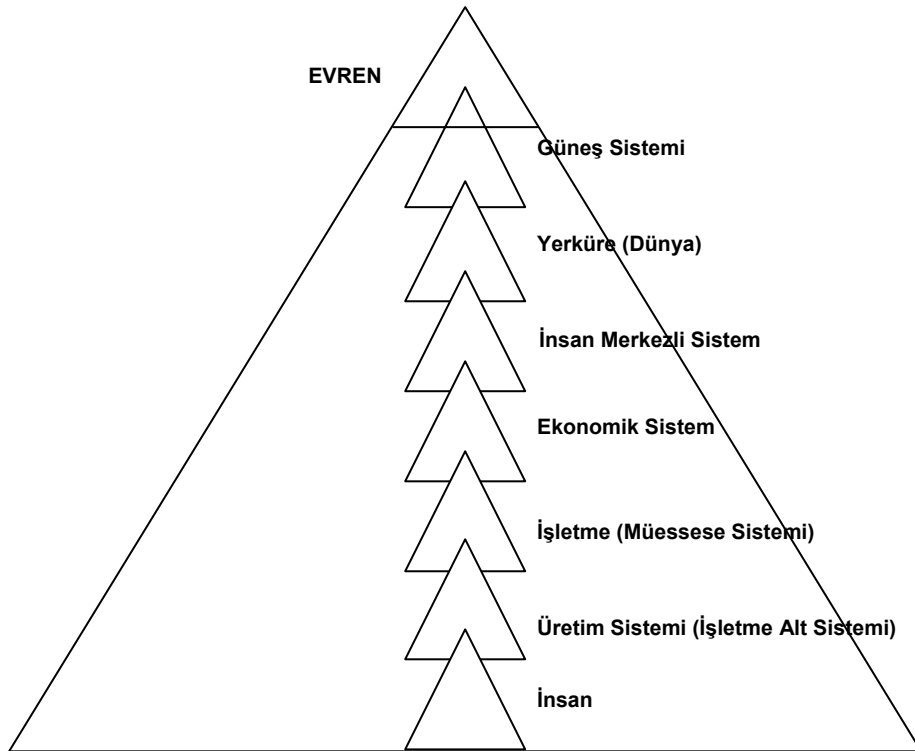
Şekil 1.3 – Geri beslemeli sistem (dinamik sistem)



Şekil 1.4 – Detaylı Sistem Gösterimi

#### 1.4. Sistem Hiyerarşisi

Varolan tüm sistemleri barındıran ve piramit şeklinde gösterilebilecek bir sistemler hiyerarşisinden söz etmek mümkündür. Bu hiyerarşi aşağıdaki şekilde incelenebilir. Burada amaç bir işletme sisteminin tüm sistemler içindeki yerinin gösterilebilmesidir.



Şekil 1.5 – İşletme Açısından Sistem Hiyerarşisi

#### 1.5. Sistem Sınıflandırması

Sistemleri farklı şekillerde sınıflandırmak mümkündür: (1) açık ve kapalı sistemler, (2) canlı ve cansız sistemler, (3) doğal ve insan yapısı sistemler, (4) statik ve dinamik sistemler, (5) soyut ve somut sistemler, (6) basit ve karmaşık sistemler. Bu sınıflandırmalar aşağıda açıklanmıştır.

### 1.5.1 Açık ve kapalı sistemler

Kapalı sistemler, çevreyle etkileşimi olmayan sistemlerdir. Aslında çevresiyle hiç bir şekilde girdi-çıkı alışıverişinde bulunmayan bir sistem örneği bulmak hemen hemen imkansız olduğundan dolayı bu tür sistemler genelde teorik ve varsayıma dayalı sistemlerdir. Açık sistemlerde çevre ile sistem arasında bilgi, malzeme ve enerji değişimi vardır. Bazı kimyasal reaksiyonlar (nükleer reaksiyonlar gibi) kapalı sistem olarak düşünülebilir.

### 1.5.2 Canlı ve cansız sistemler

Doğum, ölüm ve çoğalma gibi biyolojik özelliklere sahip sistemlere “canlı sistemler” denir. Biyolojik bir yaşam belirtisi göstermeyen sistemler ise cansız sistemlerdir. Bir insan ya da hayvan canlı sistemler için örnek oluştururken, bir uçak ya da bir müessese cansız sistemlere örnektir.

### 1.5.3 Doğal ve insan yapısı sistemler

Adından da anlaşılacağı gibi insanlar tarafından belli amaçlar doğrultusunda meydana getirilen sistemler insan yapısı sistemlerdir (artificial systems). Bunun tersi doğal yollarla oluşmuş olan sistemler doğal sistemlerdir. Bir işletme ya da işletmeyi de içine alan ekonomik sistem insan yapısı bir sistemdir. Güneş sistemi ya da dünyamızdaki tabii hayat ise doğal bir sistemdir.

### 1.5.4 Statik ve dinamik sistemler

Çevredekileri değişimlere karşın durumunu koruyan sistemler statik sistem olarak adlandırılırken, çevredeki değişikliklere göre zaman içinde değişikliğe uğrayan sistemlere dinamik sistemler denir. Dinamik sistemler bir geri besleme mekanizması sayesinde kendisini çevredeki değişken parametrelere uydurur. Statik sistemlerse uzun müddet durumlarını korurlar. Örneğin bir işletme, çevredeki arz ve talep gibi değişken parametrelerin zaman içindeki durumuna göre kendisini sürekli değiştirmek ve ayarlamak durumunda olduğu için dinamik bir sistemdir. Güneş sistemimiz ise bizim zaman ölçeğimiz içinde düşünüldüğünde hemen hemen hiçbir değişikliğe uğramadan seyrini sürdürmektedir. Güneş sistemi bu açıdan statik sistemlere örnek olarak verilebilir.

### 1.5.5 Soyut ve somut sistemler

Eğer bir sistem somut öğelerden meydana geliyorsa o sistem bir somut sistemdir. Tüm elemanları kavramlardan oluşan sistemler ise soyut sistemlerdir. Buna göre somut bir sistem kavramlardan ve fiziksel nesnelere oluşuyor olabilir. Akla ilk etapta gelen sistemlerin hemen hepsi somut sistemlerdir; işletme sistemi gibi. Soyut sistemlere örnek olarak ise basit bir bilgisayar programı verilebilir. Soyut sistemler için bir diğer örnek de felsefe sistemi olabilir.

### 1.5.6 Basit ve karmaşık sistemler

Sistemde çok az öğe ve ilişki varsa basit sistemdir. Örneğin bir çörek pişirme işlemi basit bir sistemdir. Karmaşık sistemler ise çok fazla öğe ve ilişki barındıran sistemlerdir. Makine imalatı yapan bir işletme karmaşık bir sistem sayılabilir.

## 1.6. Endüstri Mühendisliği ve Sistem Yaklaşımı

Mühendislik, bir takım faydalı şeyler yaratmak veya inşa etmek amacıyla onların tasarımı ile ilgilidir. Günümüzde mühendislik sadece insanlığın faydası için bir takım şeyleri tasarlamakla kalmayıp aynı zamanda onların, maliyetlerini de dikkate alarak daha etkin bir şekilde faaliyette bulunmasını sağlamaya çalışır. Yaratıcılık, etkinliği artırmaya yönelik tasarım ve kaynakların ekonomik kullanımı, mühendisliğin temel ilkeleridir.

Mühendislik mesleği, tasarım problemlerinin ele alınmasında sistematik, bilimsel ve matematiksel yaklaşımlar kullanır.

İnsanlar tarafından meydana getirilen sistemlerin elamanları insanlar, araçlar, makineler, hammaddeler ve diğer fiziksel nesnelere dir. Şimdi bu bilgiler ışığında Endüstri Mühendisliğinin tanımına bir bakalım:

**Endüstri mühendisliği;** insan, hammadde ve diğer teçhizatın tasarımı, geliştirilmesi ve düzenlenmesi ile ilgilidir. Bu mühendislik dalı, insan, hammadde ve teçhizattan oluşan sistemlerden elde edilebilecek sonuçları belirlemek, kestirmek ve değerlendirmek için özel bir ihtisas alanı olan mühendislik yaklaşımının analiz ve tasarım ilke ve yöntemlerini kullanır.

Bu tanımdan da anlaşılacağı gibi endüstri mühendisliği insan-makine ve diğer bileşenlerin oluşturduğu sistemleri bilimsel yaklaşımla ele almaktadır. Bu nedenle sistem teorisi ile yakın ilişkisi vardır. Endüstri mühendisleri görev yaptıkları kurumları bir sistem olarak kabul eder ve sistem yaklaşımının gerektirdiği şekilde o kurumların işleyişine katkıda bulunmaya çalışırlar.

## 2. SİSTEM MODELLERİ

Sistemlerin işleyişini ve durumlarını izah etmek ve göstermek amacıyla çeşitli modellerden faydalanılır. Bu modeller aşağıdaki şekilde sınıflandırılabilir:

### 2.1. Sözlü (Kavramsal) Modeller

Sistem modelleri içinde en eski ve en genel olanı sözlü, diğer bir ifadeyle de kavramsal modellerdir. Bu modeller, sistemi sözcüklerle açıklamaya çalışırlar. Bu modellerin avantajları, düşük maliyetli olmaları, kolay kurulabilir olmaları ve karmaşık olmayan sistemlerde kolay anlaşılabilir olmalarıdır. Ancak sözcüklerin kullanıldığı durumlarda, farklı insanlar sözcüklere birbirlerinden farklı anlam yükleyebildiklerinden yanlış anlaşılmalara karşılaşılabılır.

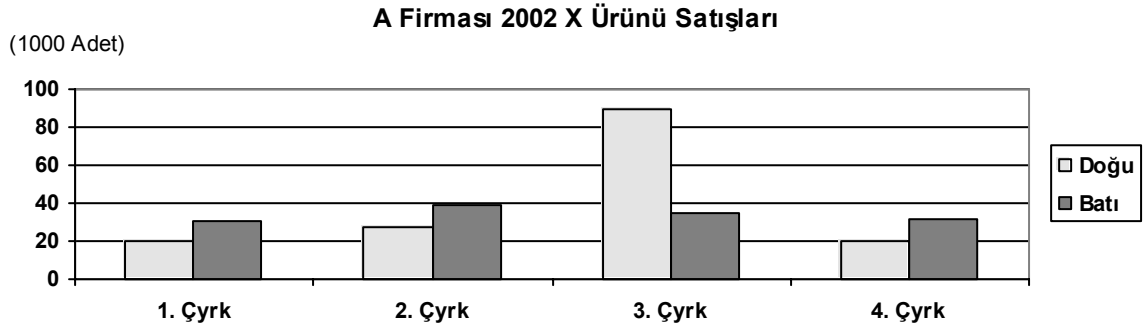
### 2.2. Şematik Modeller

İnsanların bilgileri gözle görerek algılama kabiliyetleri oldukça yüksektir. Bu nedenle doğru tekniklerle oluşturulmuş şekillerle anlatılan bilgileri daha kolay ve çabuk kavrayabilirler. Sistem modellemede de şematik modellerin kullanımı yanlış anlamaları önlemek açısından önemlidir.

Sistem modelleme de kullanılacak bazı şematik model teknikleri şunlar olabilir:

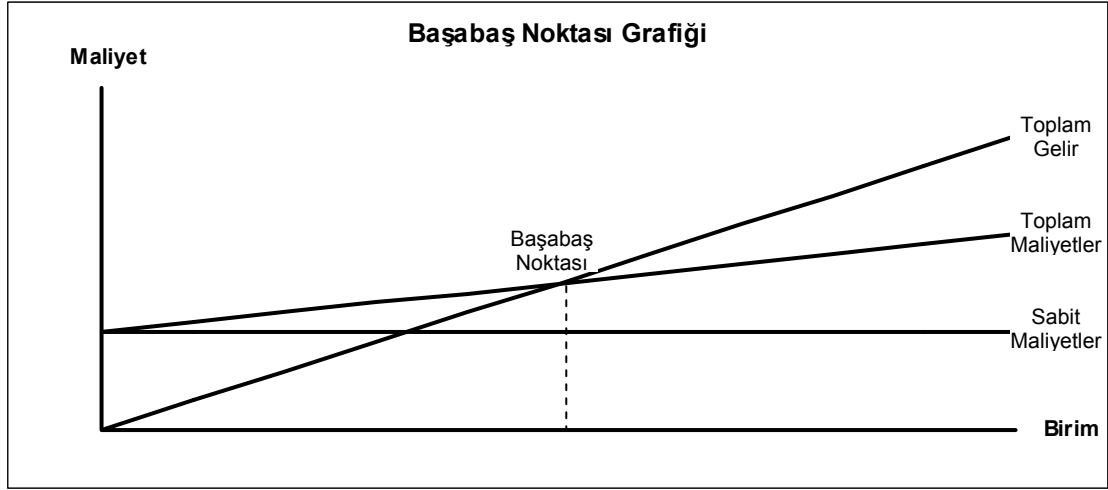
#### 2.2.1 Grafikler

Sistemin belirli parametreler açısından zamanın bir anındaki ya da zaman içerisindeki durumunu göstermek için grafikler kullanılabilir. Örneğin, bir firmada yıl içinde yapılan satışlar aylık çubuk grafik yardımıyla gösterilebilir (Şekil 2.1).



Şekil 2.1 – Bir Çubuk Grafik Örneği

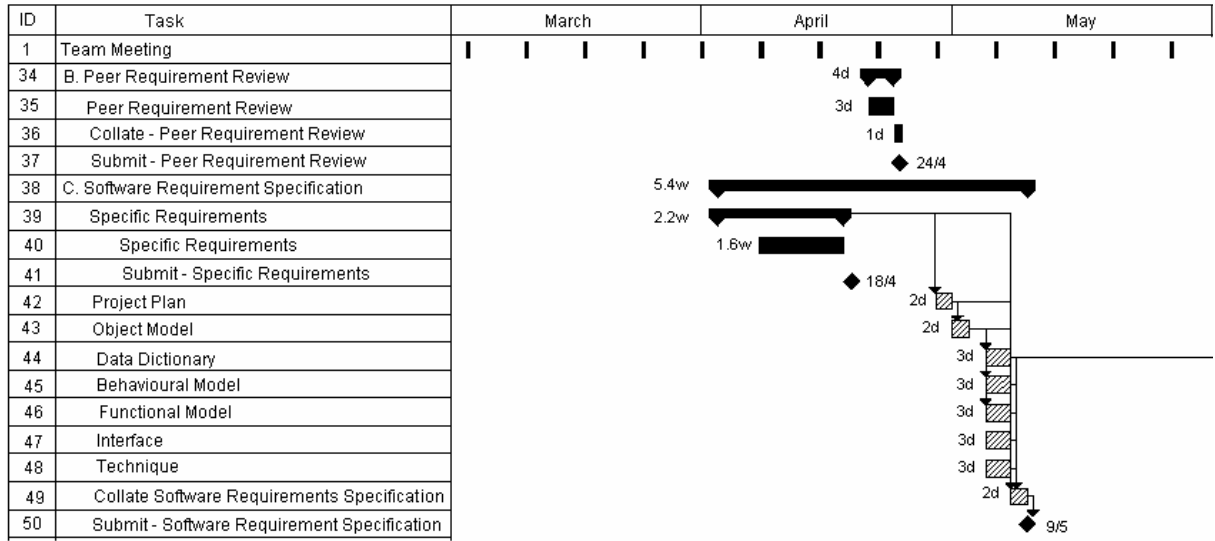
İşletme sistemi tarafından üretilen bir ürünün maliyet incelemesi de bir grafik yardımıyla sunuluyor olabilir. Böyle bir grafik yardımıyla ürünün sabit ve değişken maliyetlerinin yanı sıra, başa baş noktası gösterimi ile fiyat analizi de yapılabilir.



Şekil 2.2 – Başabaş Noktası Grafiği

### 2.2.2 Gannt Şeması

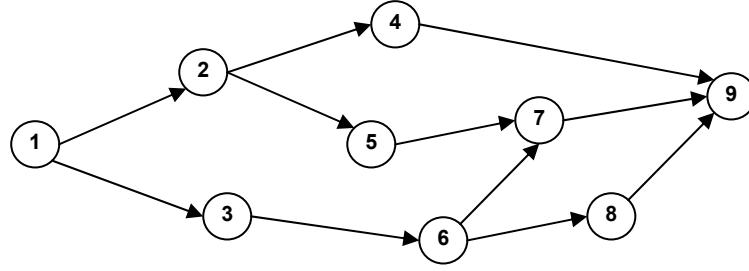
Proje yönetimi tekniğinin önemli tekniklerinden biri sayılan ve bir proje kapsamında yapılması gereken işleri gösteren Gannt şeması da sistem içindeki işleyişi göstermek için kullanılabilir. Bu sistem içerisinde yapılacak olan işlemleri bir zaman planında göstermek için kullanılan bir araçtır. Bir örnek Gannt şeması aşağıda verilmiştir.



Şekil 2.3 – Gannt Şeması Örneği

### 2.2.3 Ağ Diyagramı

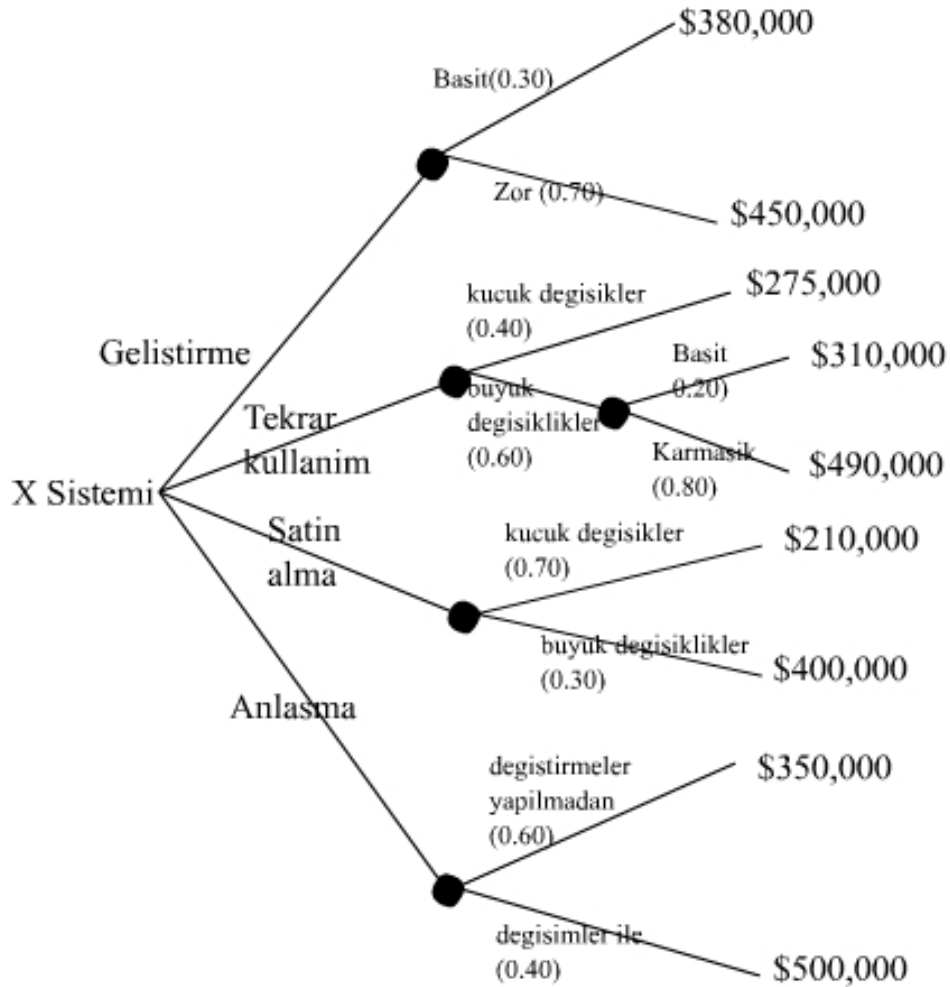
Sistemdeki bazı faaliyetlerin çözümlenmesi ve optimal sonuçlara ulaşılabilmesi için ağ diyagramlarından faydalanabilir. Örneğin bir içecek fabrikası dağıtım sisteminde kamyonların hangi rotayı izleyeceğini tespit etmek için bir ağ diyagramından faydalanabilir. Aşağıdaki şekilde örnek bir ağ diyagramı verilmiştir.



Şekil 2.4 – Bir Ağ Diyagramı Örneği

### 2.2.4 Karar Ağacı

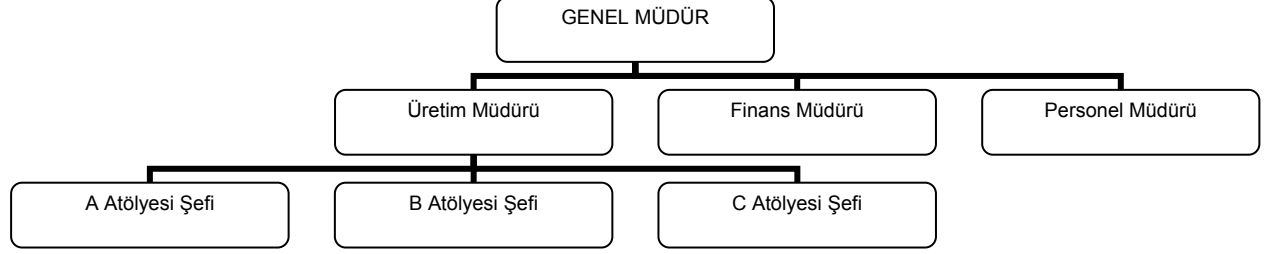
İşletmelerde sistemle ilgili kararlar verilirken, alınan kararların sistemi götüreceği sonuçları kestirmek için karar ağaçları kullanılabilir. Deterministik ya da stokastik karar ağaçlarını kullanmak mümkündür. Stokastik karar ağaçlarında oluşacak sonuçlar açısından olasılıklar da hesaplanabilir. Örnek bir karar ağacı aşağıda verilmiştir. Bu örnekte mevcut eskimiş bir yazılım sistemi için alternatifler değerlendirilmekte ve alınacak karara göre oluşabilecek maliyetler olasılıklı olarak hesaplanmaya çalışılmaktadır.



Şekil 2.5 – Bir Karar Ağacı Örneği

## 2.2.5 Organizasyon Şeması

Bir işletme sistemindeki hiyerarşiyi göstermek için kullanılırlar ve işletme sisteminde nasıl bir örgütlenme olduğunun iyi bir göstergesidirler. Aşağıda bununla ilgili basit bir örnek sunulmuştur:



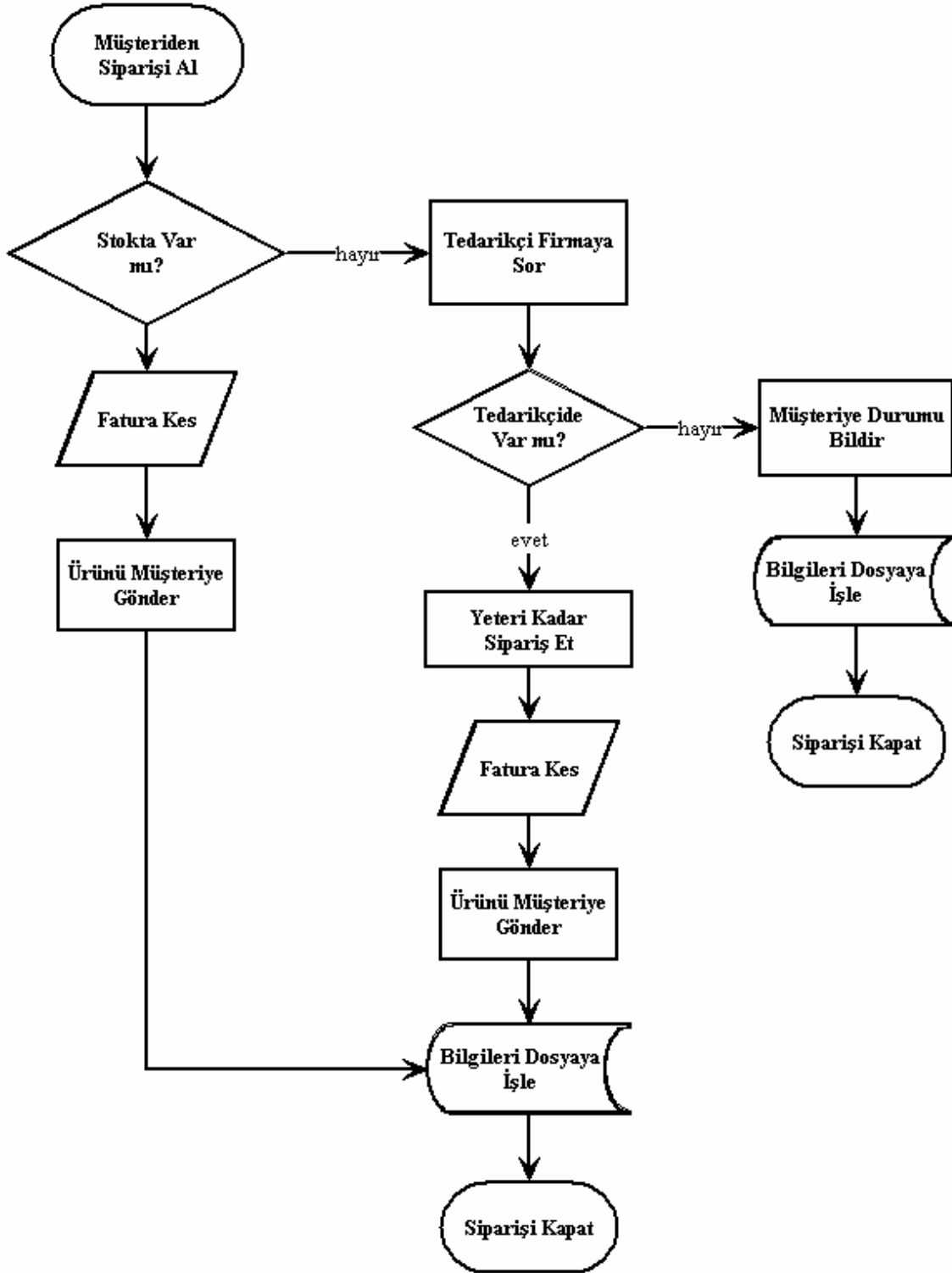
Şekil 2.6 – Bir Organizasyon Şeması Örneği

## 2.2.6 Süreç Akış Şeması

Sistemde bulunan genel sürecin (proses) yada alt süreçlerin nasıl işlediğini izah etmek için kullanılan şematik bir gösterimdir. Süreç akış şeması için kullanılan şekiller ve açıklamaları ile örnek bir süreç akış şeması aşağıda verilmiştir. NCC (National Computing Centre – İngiltere) tarafından geliştirilen bu simgeler bilgisayar programlarının akış diyagramları için de kullanılır.

Şekil	Açıklaması	
	Sistem Akış	Bilg. Program Akış
	Başlangıç / Bitiş	Başlangıç / Bitiş
	İşlem	İşlem
	Karar	Karar
	Doküman Belge	Girdi / Çıktı
	Bilgi Depolama (Dosyalama)	Bilgi depolama
	Kullanılmaz	Önceden tanımlanmış işlem
	Sayfa İçi Bağlayıcı	Sayfa İçi Bağlayıcı
	Sayfalar Arası Bağlayıcı	Sayfalar Arası Bağlayıcı

Tablo 2.1 – Süreç Akış Şeması Sembolleri (NCC)



Şekil 2.7 – Bir Süreç Akış Şeması Örneği (Telefonla Satış)

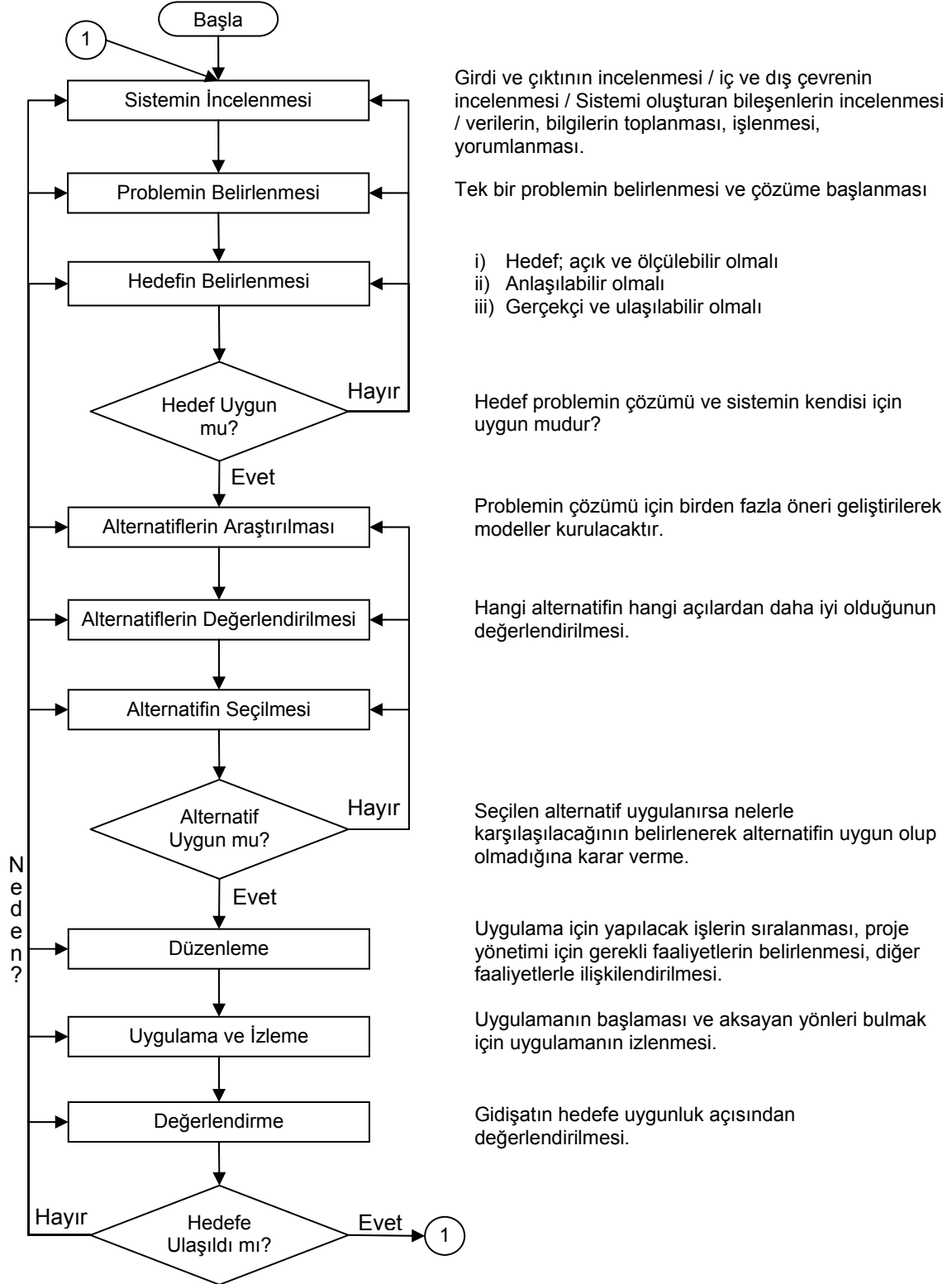
**Ödev 1:** Aşağıda tanımlanan sistemin süreç akış diyagramını sağlık bakanlığı görevlisinin yapacağı işler açısından gösterecek şekilde bir A4 kağıt üzerine bilgisayar ile çizin ve bir sonraki derste teslim edin.

Bir doktorun (hekim), özel muayenehane (ofis) açabilmesi için sağlık bakanlığından izin alması gerekmektedir. Bunun için bakanlık görevlisi, başvuru formunu ve gerekli diğer belgeleri başvuru sahibinden alıp kontrol etmekte ve belgeler eksikse tamamlanması için başvuru sahibine iade etmektedir. Eğer belgeler tamam ise geçici izin belgesini doldurarak başvuru sahibine teslim etmekte ardından da aldığı belgeleri dosyalamaktadır. Daha sonra da bir üst birime asıl belgeyi hazırlamaları için talepte bulunmakta ve belge dosyasını bu üst birime göndermektedir.

### 3. SİSTEM ANALİZİ

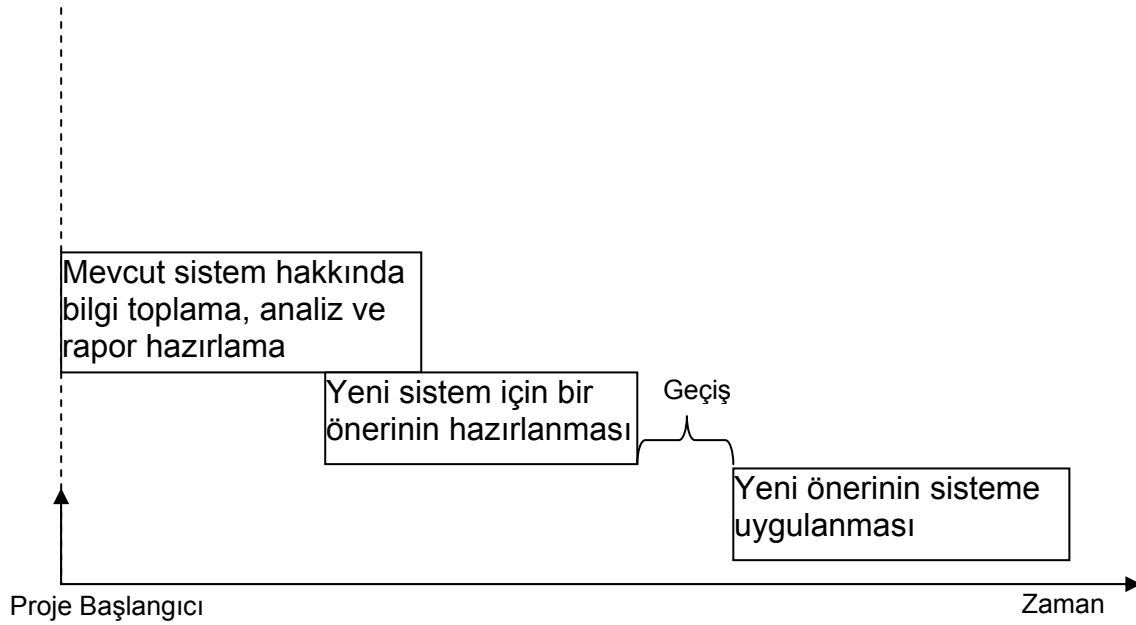
Bu bölümde sistem analizi aşamaları ile yeni sistemlerin geçiş yöntemleri anlatılacaktır.

#### 3.1. Sistem Analizi Aşamaları



Şekil 3.1 – Sistem Analizi Aşamaları

### 3.2. Sistem Analizinin Temel Faaliyetleri (Gantt Şeması)

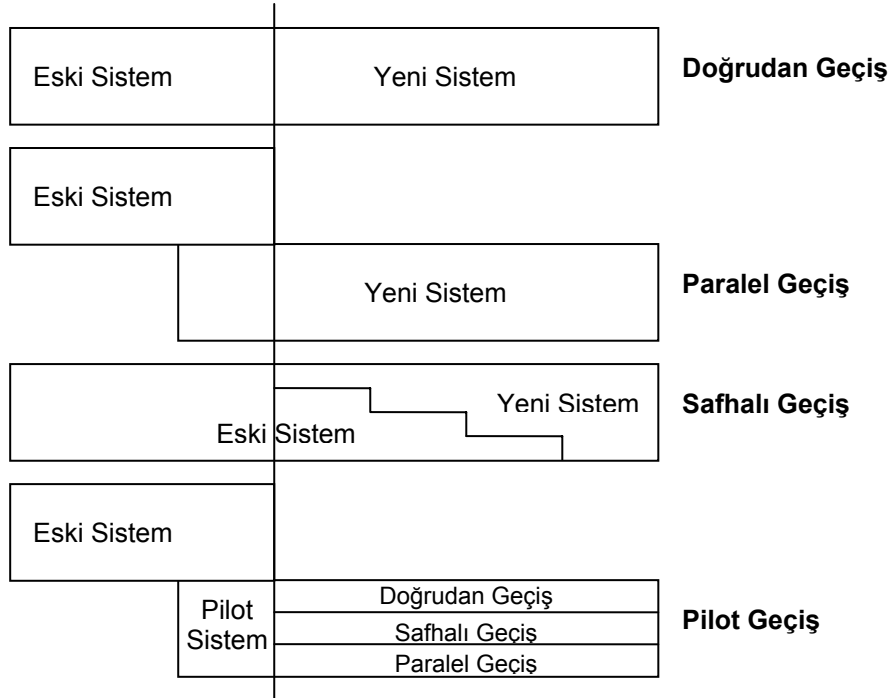


Şekil 3.2 – Sistem Analizi Faaliyetleri

Şekil 3.2’de sistem analizi faaliyetleri bir Gantt şeması üzerinde gösterilmiştir. Buna göre yeni tasarımla eski tasarım arasında bir geçiş yaşanmaktadır.

### 3.3. Yeni Sisteme Geçiş Yaklaşımları

Dört adet geçiş yaklaşımı vardır. Bu geçiş yaklaşımları Şekil 3.3’te gösterilmiş ve bu şeklin ardından her bir geçiş tipine ilişkin açıklama verilmiştir.



Şekil 3.3 – Geçiş Yaklaşımları

### 3.3.1 Doğrudan Geçiş

Belirlenen bir günde, eski sistemden yeni sisteme doğrudan geçişi ifade etmektedir. Kurulan yeni sistem yeni bir sistemin yerine geçmiyorsa yada eski sistem artık görevini ifa edemiyor durumdaysa bu yaklaşım tercih edilebilir. Geriye dönüşü çok zor ve maliyetli olduğu için genelde küçük firmalar tarafından tercih edilir. Riski fazladır.

### 3.3.2 Paralel Geçiş

Yeni sistemin tam olarak çalıştığı anlaşılana kadar eski sistemle yeni sistemin aynı anda paralel olarak işletilmesidir. Eğer yeni sistem, eskiden çalışmakta olan ve istenildiği kadar olmasa da verim sağlayan bir sistemin yerine tasarlanmışsa, bu durumda iki sistemin bir müddet birlikte çalışmasında fayda vardır. Eski sistem ile yeni sistemi kıyaslama şansı verir. Yeni sistemin istenildiği gibi çalışmaması durumunda eski sisteme dönüşe müsaade ettiği için riski yüksek değildir. Buna karşılık aynı iş için iki ayrı sistem aynı anda kullanıldığı için maliyeti yüksektir. Yeni sistemin yeterli olduğuna kanaat getirildiğinde eski sistemin uygulamasına son verilir.

### 3.3.3 Safhalı (adım adım) Geçiş

Yeni sistemin, parça parça uygulamaya konulmasıdır. Örneğin, bir satış bilgi sisteminde, ilk önce satışların muhasebelenmesi modülü, daha sonra stok yönetimi modülü vb. Uygulamaya konabilir. Buna göre alt sistemlerden biri yeni sisteme geçerken diğer alt sistemler yapılan plana göre bir müddet daha işlemeye devam etmektedir. Büyük ölçekli sistemler için tercih edilen bir yöntemdir. Dezavantajı geçiş zamanının uzun vadeye yayılmasıdır.

### 3.3.4 Pilot Geçiş

Pilot, komple çalışma sisteminin bir alt kümesinde yürütülen bir deneme sistemidir. Yeni sisteme geçiş bu şekilde bir pilot uygulama ile gerçekleştirilebilir. Örneğin yeni bir müessese bir üretim sistemini 8 fabrikada uygulayacaksa önce bu fabrikalardan birini pilot olarak seçip sistemi o fabrikada deneyebilir. Pilot yürütülürken genelde eski sistem muhafaza edilmekte fakat aktif olmamaktadır. Pilot sistem başarıya ulaşırsa diğer fabrikalara da aynı sistem kurulur.

## 4. BİLGİ SİSTEMLERİNE GİRİŞ (YBS)

### 4.1. Veri ve Bilgi

Bilgi kelimesi, Türkçe'de birbirinden farklı iki kavrama karşılık gelmektedir. Birincisi, nesnelere veya kavramlar hakkında genel olarak sahip olunan görüş ve ikincisi nesnelere ya da kavramların çeşitli niteliklerini tanımlayan işlenmiş veri. Birinci türdeki bilgi, sistemlerin ve çevrelerinin işleyişini düzenleyen kuralların bilgisidir. Örneğin, bir doğrusal karar modelinin simpleks algoritması yardımıyla nasıl çözüleceğini bilmek ya da nasıl araba kullanıldığını bilmek gibi. Bu tür bilgi **yöntem bilgisidir**.

İkinci tür bilgi ise **durum bilgisidir**. Bu türden bilgiler, sistemin ya da çevresinin durumunun, durumu tanımlayan değişkenlerin değerleri aracılığı ile bilinmesini sağlarlar. Örneğin, maliyetlerin sabit ve değişken olmak üzere iki sınıf altında toplandığını bilmek, bir kuruluşta değişken maliyetin ne kadar düşürülmesiyle ne kadar kâr artışı sağlanacağını bilmeye yetmez. Böyle bir sonuca ulaşabilmek için kuruluştaki mevcut sabit ve değişken maliyetlerin ne kadar olduğunu bilmek gerekir. Bu bilgiler, maliyet sisteminin durumunu gösteren durum değişken değerleridir.

Veri kavramı, genelde durum bilgisi kavramı ile karıştırılmaktadır. Oysa ikisi arasında ciddi fark vardır. Veri, gerçeklik üzerinde yapılan gözlemlerin sonucu ve bu anlamda bilginin üretildiği hammadDEDİR. Başka bir deyişle veri, kullanıcılar için herhangi bir anlam ifade etmeyen olgular ve şekillerdir. Bilgi ise, karar vermede faydalı olacak şekilde verinin dönüştürülmesi ve analiz edilmesiyle anlamlı hale getirilmesidir.

Örneğin sabah işe ya da okula giderken iki veri toplarız. Birincisi o anki mevcut zaman bir diğeri ise işte ya da okulda kaçta olmanız gerektiği. Zihnimizden bu iki zamanın farkını alırsak ne kadar vaktimiz kaldığı bilgisine erişiriz. Yani bu iki veriyi işleyerek bilgiye dönüştürmüş oluruz. Bu bilgiye dayanarak da acele etmeliyim, bugün kahvaltı etmeyeceğim gibi kararlar verebiliriz.

### 4.2. Bilginin Karakteristikleri

Bilginin, yöneticilerin karar vermesine yardımcı olması, verilen kararların belirsizliğini azaltabilmesi, yani yararlı ve değerli olabilmesi için aşağıdaki özelliklere sahip olması istenir:

- (i) **Bilginin Doğruluğu ve Doğrulanabilirliği**: Bilginin doğruluk kalitesi, onun hatadan bağımsız olma (hatasız olma) derecesine bağlıdır ve bilgi aksi ortaya konmadıkça doğru kabul edilir. Çoğu kez %95 doğru bilgiye ulaşmak ekonomik olmayabilir. Örneğin bir ürüne ait pazar araştırmasında müşterilerin ancak bir bölümüyle anket yapabilir ve bu örneği kullanarak belli bir güvenlik seviyesinde (%95 gibi) güvenlik seviyesinde tüm müşterilerin görüşünü ortaya koyabilirsiniz.
- (ii) **Bilginin Tamlığı**: Bilgi tamamen doğru ve doğrulanabilir olmasına karşın tam olmayabilir. Bir karar vericiye sağlanan bilgi miktarı ile o bilginin tamlığı arasında bir ilişki yoktur. Örneğin fayda/maliyet analizine ilişkin bir analiz yaparken fayda bilgisi tamken, maliyet bilgilerinde eksiklikler varsa bu eksiklik yanlış bir yatırım kararı alınmasına yol açabilir. Bu açıdan sağlanan bilgi tam olmalı konuyla ilgili bilinmesi gereken tüm yönleri kapsamalıdır.
- (iii) **Bilginin Zamanlılığı**: Bilgi kendisine ihtiyaç duyulduğunda hazır olmalıdır. Bilgi doğru ve tam olmasına rağmen zamanında elde edilememişse,

yönetici için çok şey ifade etmeyecektir. O bilginin ihtiyaç duyduğu karar prosesi geçmiş ve karar çoktan verilmiş olduğundan o bilgi anlamını yitirmiş olacaktır. Örneğin bir müşterinin bir ürün için talepte bulunduğunu ve bir gün içinde ürünü kimden alacaklarına karar vereceklerini ve hemen alım yapacaklarını söylediğini varsayalım. Eğer biz o gün içinde, elimizdeki mevcut stokların sayısını elde edemezsek müşteriye gün içinde istediği cevabı verememiş oluruz. Bu bilginin ertesi gün elimizde olması ise pek bir şey ifade etmeyecektir.

- (iv) **Bilginin İlgililiği:** Bilginin ilgililik kalitesi, belirli bir kararda, bilginin girdi olarak ilgili olmasına bağlıdır. Yani bilgi, karar vericinin karar vereceği konu ya da konularla ilgili olmalıdır. Eğer bir restorandaki rezervasyon sistemiyle ilgili bir karar vermeye çalışıyorsak restorandaki boş yer sayısı ilgili bir bilgidir, ancak o akşamki menüde hangi yemeklerin olduğu bilgisi bu karar açısından yeterince ilgili bir bilgi değildir.
- (v) **Bilginin ekonomikliği:** Gerçek durum tam olarak bilinmese de, bilgi sağlamanın belirli bir maliyeti vardır. Karar vericiler sürekli olarak, bilginin üretilme maliyeti ile sağladığı fayda arasında bir denge oluşturmak zorundadırlar.

### 4.3. Yönetim ve Karar Verme Seviyeleri

Karar verme seviyeleri 3 şekilde sınıflandırılabilir. Başka bir ifadeyle, 3 yönetim seviyesi bulunmaktadır.

- (i) **Stratejik Karar Verme:** Üst seviye yöneticilerin verdiği kararlardır. Geleceğe yöneliktir ve bu kararların belirsizlik seviyesi oldukça yüksektir. Stratejik karar verme, organizasyonun amaçlarının belirlenmesi ve bu amaçlara ulaşmak için uzun dönem planların yapılmasını içerir. Örneğin, yeni üretim tesisi inşa edilmesi, hangi ürünlerin üretileceği ile ilgili kararlar gibi.
- (ii) **Taktik Karar Verme:** Orta seviye yöneticilerin verdiği kararlardır. Stratejik seviyede verilen kararların yerine getirilmesinde, kaynakların etkin ve verimli olarak elde edilmesi ve kullanılmasına yöneliktir. Organizasyonel amaçları yerine getirmek için kaynakların tahsis edilmesini içerir. Örnek olarak, tesis yerleşimi, bütçe tahsisi ve üretim planlama gibi kararlar verilebilir.
- (iii) **Operasyonel Karar Verme:** Alt seviye yöneticilerin verdiği kararlardır. Taktik seviyedeki kararların yürütülmesi için gerekli görevlerin etkin ve verimli bir şekilde yapılmasını içerir. Örneğin işlerin çalışanlara tahsisi, sipariş zamanlarının belirlenmesi gibi.

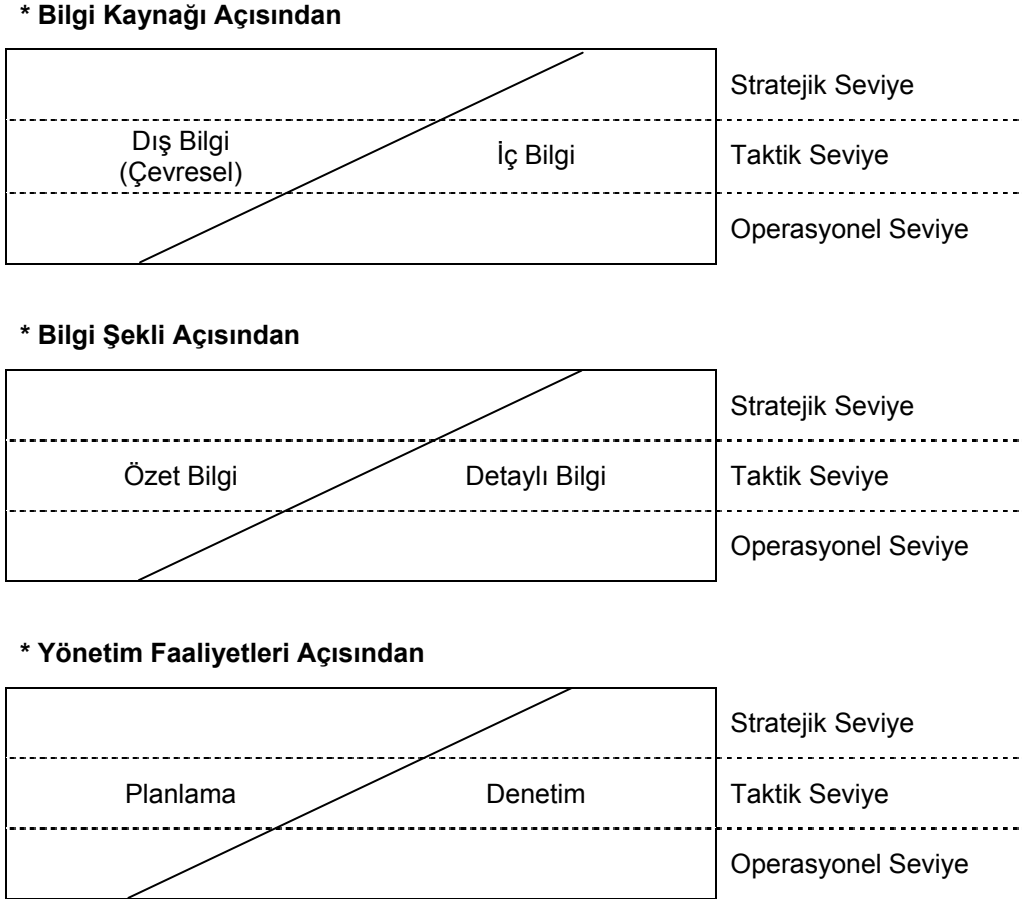


Şekil 4.1 – Karar Verme (Yönetim) Seviyeleri

Karar verme seviyelerinin ki buna yönetim seviyeleri de denir, şematik gösterimi Şekli 4.1'de görülmektedir. Şekilde de görüleceği gibi tüm seviyelerin altında veri işleme / operatör adı verilen bir seviye bulunmaktadır. Bu seviye, değişik seviyelerce verilecek olan kararlar için ihtiyaç duyulan bilgi üretimini gerçekleştirir.

#### 4.4. Yönetimde Bilgi Kullanımı

Her yönetim seviyesinde ihtiyaç duyulan bilgiler birbirinden farklıdır. Şekil 4.2'de bu farklılıklar, bilgi kaynağı, bilgi şekli ve yönetim faaliyetleri açısından incelenmiştir.



Şekil 4.2 – Yönetim Seviyelerinin Bilgi İhtiyaçları

#### 4.5. Karar Tipleri

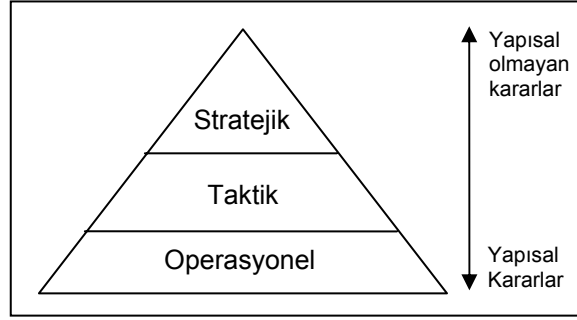
Kararlar, yapısal (programlanmış) ve yapısal olmayan (programlanmamış) kararlar olarak iki kısma ayrılırlar.

i) Yapısal Kararlar: Programlanmış kararlar olarak da bilinen bu tip kararlar problemlerin çok iyi tanımlandığı kararlardır. Bu tür kararlarda karar verici sübjektif bir esnekliğe sahip değildir. Zira gerçek kararlar, mevcut politika ve prosedürlerle belirlenmiştir. Aslında bu tip kararların çoğu insanı aracı yapmadan bilgisayar tarafından verilebilmektedir. Örneğin malzemenin stok seviyesiyle ilgili kararlar genellikle yapısal kararlardır. Bir malzemenin stok miktarının emniyet stok seviyesinin altına düştüğünde yeniden sipariş edilmesi kararını bir bilgisayar verebilir.

ii) Yapısal Olmayan Kararlar: Programlanmamış kararlar olarak da bilinen bu tip kararlar problemlerin çok iyi tanımlanmadığı kararlardır. Bu tür kararlar bilgi

tabanlı kararlar olarak da alınırlar, zira karar verici rasyonel bir karar vermek için bilgiye ihtiyaç duyar. Bu bilgi ihtiyacı, karar verme prosesinde yöneticinin yargı ve sezgisini kullanma ihtiyacını ortaya çıkarır. Başka bir ifadeyle yapısal olmayan kararlar, doğru cevabı bilmenin mutlak bir yolunun olmadığı ve ortada birçok doğru karar olabildiği durumlarda söz konusu olan kararlardır. Hiçbir kriter ya da kural, iyi bir çözümü garanti etmez. Piyasaya yeni bir ürünün sürülüp sürülmeyeceğine dair karar, yeni bir pazarlama kampanyasına girilmesi, şirketi imajının değiştirilmesi gibi kararlar bu tip kararlardır.

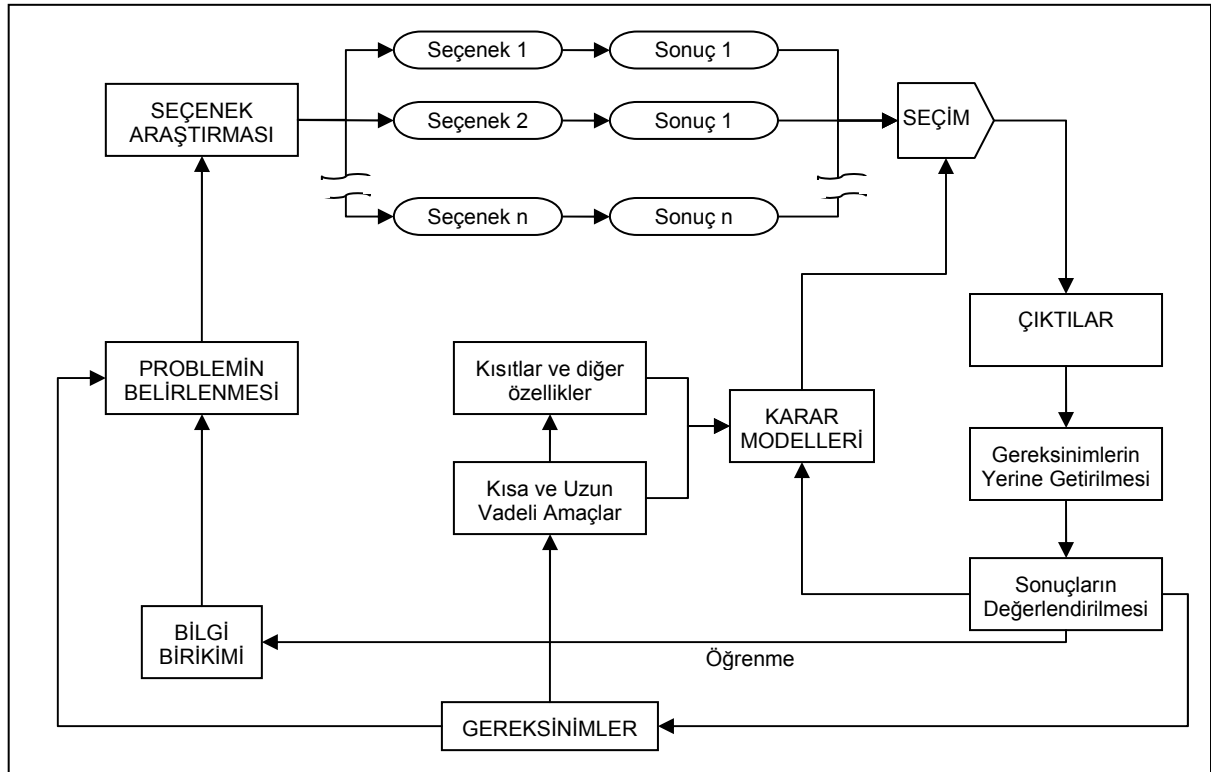
Şekil 4.3'te farklı yönetim seviyelerinde kullanılan karar tipleri gösterilmiştir.



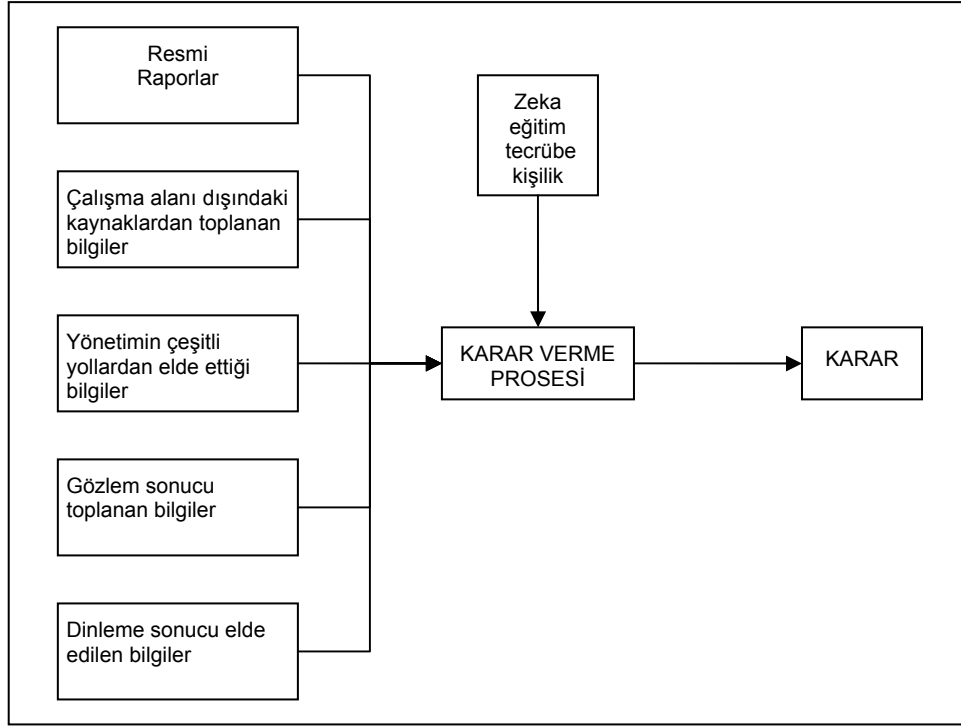
Şekil 4.3 – Yönetim Seviyelerinde Karar Tipleri

#### 4.6. Karar Verme Prosesi

Aşağıda verilen iki şekilde (Şekil 4.4 ve 4.5) genel olarak karar verme prosesi ve bir yöneticinin karar verme prosesi gösterilmiştir.



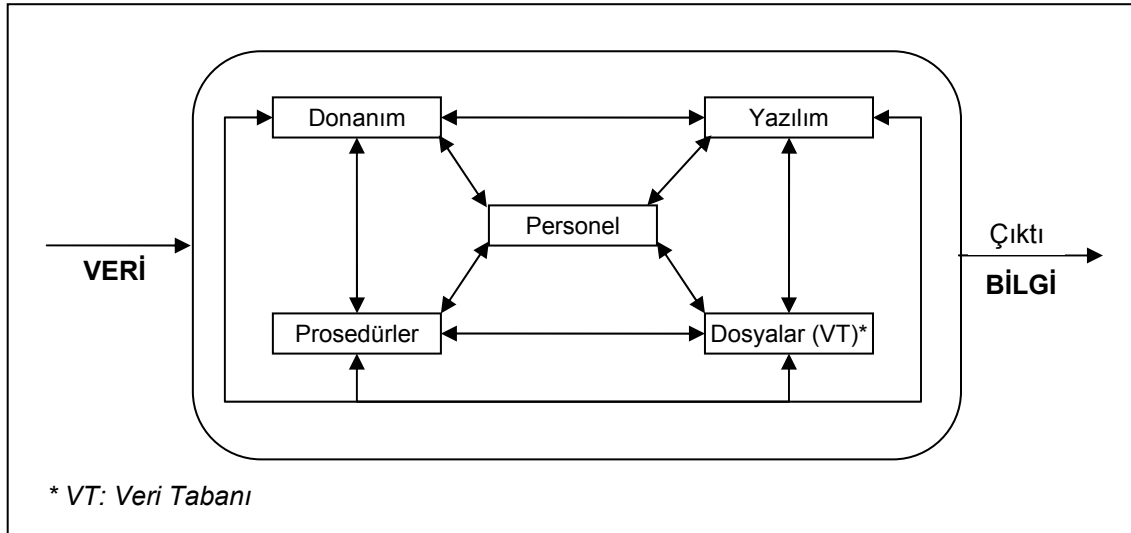
Şekil 4.4 – Karar Verme Prosesi



Şekil 4.5 – Yöneticinin Karar Verme Prosesi

#### 4.7. Bilgi Sistemleri

Bilgi sistemi, karar vericiler için verileri işleyerek bilgi sağlayan çoğunlukla bilgisayara dayalı sistemlerdir. Bilgi sistemleri yapay sistemlerdir ve karar verme prosesine yardımcı olmak amacıyla tasarlanmışlardır. Bilgi sistemleri teorik olarak manuel olabilse de artık günümüzde bilgi sistemleri bilgisayara dayalıdır. Şekil 4.6'da bilgisayara bağlı bir bilgi sisteminin öğeleri ve birbirleriyle ilişkileri gösterilmiştir.



Şekil 4.6 – Bilgi Sistemi Öğeleri ve İlişkileri

#### 4.7.1 Bilgi sistemlerinin faydaları

İyi tasarlanmış etkin bir bilgi sistemin kazandıracığı bazı faydalar şunlar olabilir:

- i) Daha iyi hizmet
- ii) Daha iyi güvenlik
- iii) Rekabet avantajı
- iv) Daha az hata
- v) Büyük ölçüde doğruluk
- vi) Yüksek kalitede çıktılar (ürünler)
- vii) Sağlıklı haberleşme
- viii) Etkinliğin artması
- ix) Verimliliğin artması
- x) Daha etkin yönetim
- xi) Daha fazla fırsatlar
- xii) İşgücü ihtiyacının azaltılması
- xiii) Maliyetlerin azaltılması
- xiv) Daha etkin finansal kararlar verebilme
- xv) Aşırı faaliyetlerin daha etkin kontrolü
- xvi) Daha etkin yönetsel karar verme

## 5. BİLGİSAYARA DAYALI BİLGİ SİSTEMLERİ

Bilgi sistemleri denildiğinde genelde algılanan bilgisayara dayalı bilgi sistemleridir. Bu dersin konusu olarak da bundan sonra bilgisayara dayalı bilgi sistemleri anlatılacaktır.

Bilgisayara bağlı bilgi sistemleri şunlardır:

- i) Kayıt/Veri İşleme Sistemleri (VİS) (Transaction/Data Processing Systems)
- ii) Yönetim Bilgi Sistemleri (YBS) (Management Information Systems)
- iii) Karar Destek Sistemleri (KDS) (Decision Support Systems)
- iv) Ofis Otomasyon/Bilgi Sistemleri (OOS) (Office Automated/Information Systems)
- v) Üst Yönetim Destek Sistemleri (ÜDS) (Executive Support Systems)
- vi) Yapay Zeka ve Uzman Sistemler (YZ ve US) (Artificial Intelligence and Expert Systems)

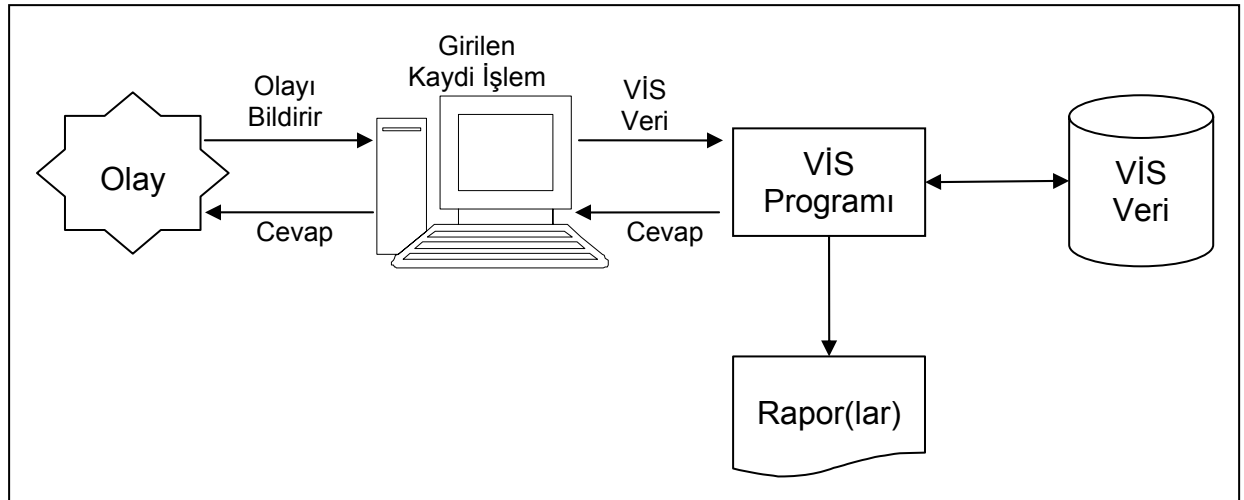
Devam eden kısımda yukarıda sıralanan bilgi sistemleri hakkında detaylı bilgi sunulmuştur:

### 5.1. Kayıt/Veri İşleme Sistemleri (VİS)

Bir VİS, işin yapılması için gerekli günlük rutin muameleleri (transaction) işleyen ve kaydeden bilgisayara dayalı sistemdir. VİS, organizasyonun operasyonel seviyesine hizmet verir. Bu seviyede, görevler, kaynaklar ve amaçlar önceden tanımlanmış kriterlere göre, düşük seviye bir yönetici tarafından verilebilir. Örneğin bir banka için bir müşteriye araç kredisi verme kararı, tüm kriterler belirlenmiş olacağı için düşük seviye bir yönetici tarafından verilebilir.

VİS, günlük operasyonlarla ilgilenir. Yapılan işlemler, işlem yükü ve hacmi çok yüksek olan tekrarlı işlemlerdir ve bu işlemlerin nitelikleri çok nadir olarak değişir.

VİS, verinin saklanması ve çağrılmasına yöneliktir ve bu özelliğiyle asıl konumuz olan YBS'nin destekleyicisi durumundadır. Bir VİS'in genel işleyişi Şekil 5.1'de gösterilmiştir.



Şekil 5.1 – Bir VİS Uygulamasının Yapısı

VİS programı iki tip çıktı üretir:

- i) Operatör terminaline gönderilen mesaj (soft copy)
- ii) Basılmış dokümanlar (hard copy)

Örneğin bilet rezervasyon sistemi için hazırlanan bir program, terminal üzerinde belirli bir kişiye hangi koltukların satıldığını gösterebilir (soft copy) yada bilet basabilir (hard copy).

VİS aşağıdaki temel özelliklere sahiptir:

- i) Kaydi işlemlerin elde edilip, kayıtların muhafaza edilmesine yöneliktir.
- ii) Dosya kökenlidir.
- iii) Çıktısı genellikle periyodiktir.
- iv) Öncelikle operasyonel seviye yönetim için bilgi üretir.
- v) Yöneticinin özel bilgi istekleri için, sınırlı esnekliğe sahiptir.
- vi) Bu sistemler tipik olarak fonksiyona dayalıdır. Uygulamalar birbirinden bağımsız olarak geliştirilir.

VİS’nde veri iki şekilde işlenir:

**1. Doğrudan (Hemen) İşleme:** Muameleler (kaydi işlemler), oluştukları anda girilir ve kaydedilir. Veritabanı üzerindeki güncellemeler hemen yapılır. Bu durumda operatör sistemden her an güncel sorgulamalar yapılabilir. Rezervasyon işlemleri, banka işlemleri gibi. (real-time processing)

**2. Parti İşleme:** bu tip işlemede muameleler gruplanır ya da kümelenir ve ardı ardına girişleri yapılır. Parti işlemede yapılan sorgulamalarda veritabanı güncel olmadığından (işlemler günlük, haftalık, aylık yapıyor olabilir) yanlış bilgiler alınabilir. Örneğin, posta işleme ve telefonla sipariş gibi uygulamalarda bu tür işleme kullanılabilir. (batch processing)

## 5.2. Yönetim Bilgi Sistemleri (YBS)

**YBS**, bir örgütün yönetiminde kullanılan bilgilerin işlenmesini ve iletilmesini sağlayan bir sistemdir.

YBS, zaman içerisinde VİS’in yetersiz kaldığı noktaları kapatmak amacıyla geliştirilmiş daha kapsamlı sistemlerdir. YBS’nin genel özellikleri şunlardır:

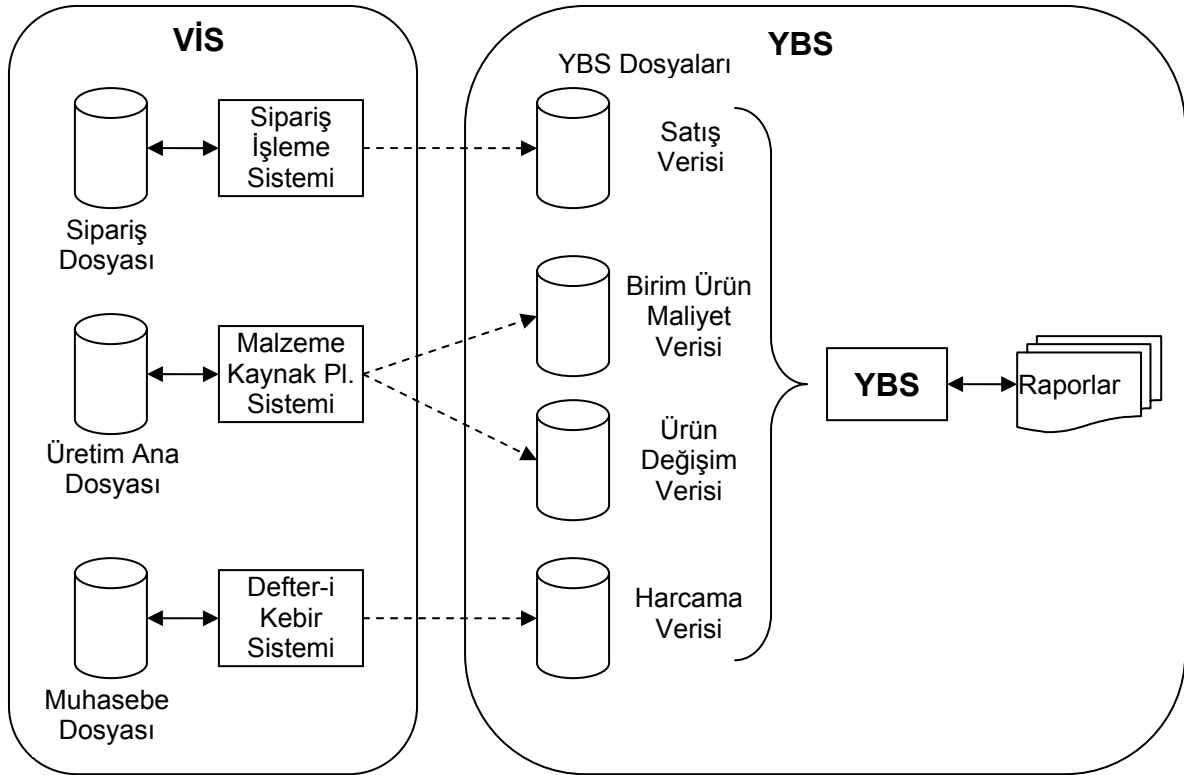
- YBS, Veri/Kayıt işleme fonksiyonlarını destekler (kayıt saklama vb).
- YBS, bütünleşik bir veritabanı kullanır ve fonksiyonel alanların çeşitliliğini destekler.
- YBS, operasyonel, taktik, ve stratejik seviye yöneticilerin bilgiye kolay ve zamanında erişimini sağlar. Özellikle yoğun olarak taktik seviye yöneticiler için hizmet sağlar.
- YBS, kısmen esnektir ve organizasyonun bilgi ihtiyaçlarındaki değişmeye adapte edilebilir.
- YBS, sadece yetkili şahısların erişimine imkan veren sistem güvenliği sağlar.
- YBS, günlük operasyonlarla ilgilenmez.
- YBS, genellikle yapısal kararların desteklenmesine yöneliktir.

- YBS, yöneticilere değişik raporlar sunar.
- YBS, öncelikle çevresel ya da dış olaylarla değil büyük ölçüde firma içi olaylara odaklanır.

### 5.2.1 YBS ve VİS'in farklılıkları

VİS, YBS için önemli bir firma içi veri kaynağıdır. Zaten, YBS genel anlamda birkaç VİS üzerine kurulmuş, örgütün ya da birkaç alt sistemin yönetsel bilgi ihtiyacını karşılamaya yönelik sistemlerden oluşurlar. VİS ve YBS arasındaki farklar şunlardır:

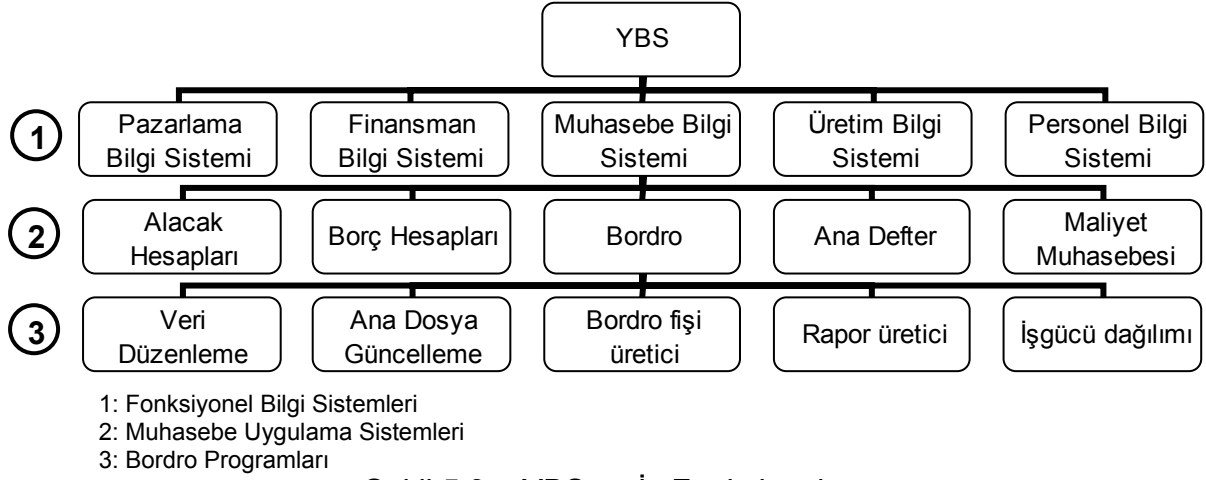
- Yöneticinin bilgi ihtiyacının karşılanmasında YBS'nin bütünleşik veritabanı, VİS'in düz dosya ortamına göre daha büyük esneklik sağlar.
- VİS, tek bir fonksiyonel alanı desteklemeye yönelmiştir, YBS ise fonksiyonel alanlar arasındaki bilgi akışını bütünleştirir.
- Bir YBS, taktik seviyeye yoğun olmakla beraber yönetimin tüm seviyelerine bilgi ihtiyaçları için hizmet sunarken, VİS sadece operasyonel seviyeye destek sağlar.
- VİS kaydi işleme yapar. VİS, bu şekilde YBS için bir veritabanı oluşturur. VİS'in çıktıları YBS için girdidir. YBS, VİS verilerini yönetimin karar vermesi için bilgi üretiminde kullanır.



Şekil 5.2 – VİS, YBS ilişkisi

Örnek: Bir bilet rezervasyon sisteminde VİS, siparişleri alma ve bilet basmada, YBS ise bilet satan her bir acentenin performansını ölçmede ve rapor etmede kullanılabilir.

### 5.2.2 YBS ve iş fonksiyonları

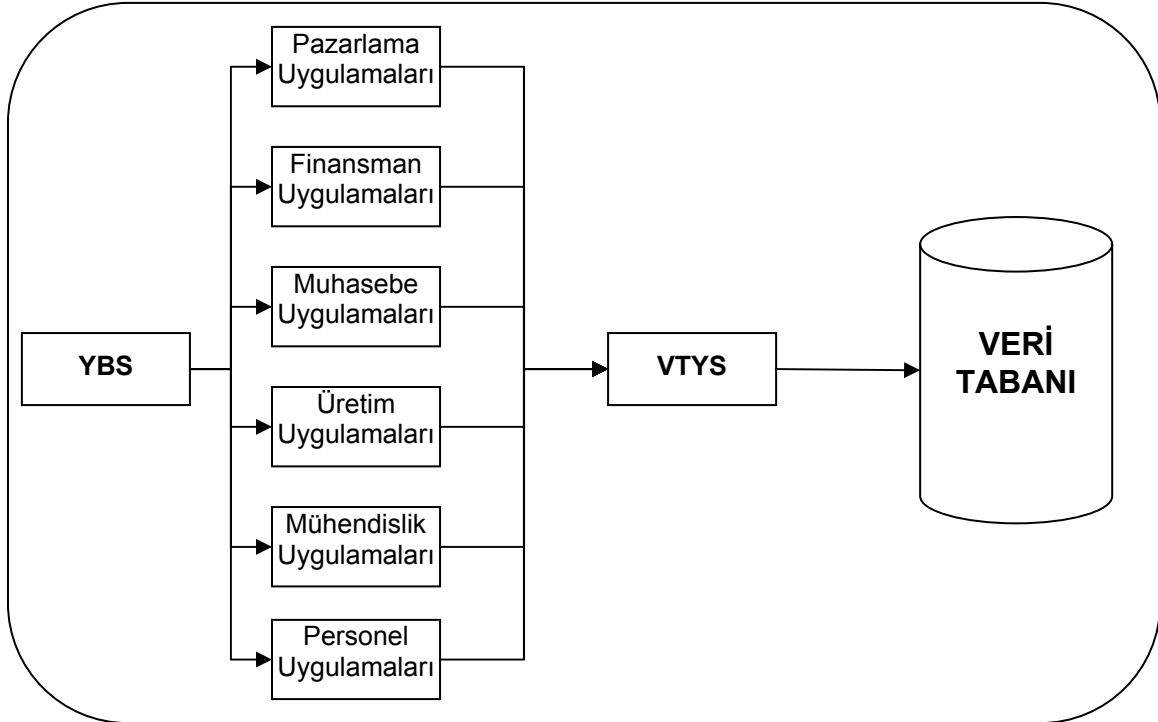


Şekil 5.3 – YBS ve İş Fonksiyonları

Şekil 5.3'te de görülebileceği gibi YBS fonksiyonel bilgi sistemlerinin birleşiminden meydana gelmiştir. Fonksiyonel bilgi sistemleri uygulama sistemlerinden onlar da programlardan meydana gelmiştir. Veritabanı Yönetim Sistemleri (VTYS), bu fonksiyonların aynı veriyi paylaşma yeteneğini artırır.

### 5.2.3 YBS ve Veritabanı Yönetim Sistemi (VTYS)

Veri, bir YBS'nin ana kaynağıdır ve bu kaynağın yönetimi oldukça önemlidir. Bir VTYS, veritabanı denen birleştirilmiş ve koordine edilmiş dosyaların kümesi ile uygulama programları arasında arayüz olarak hizmet veren bir programdır. Bu ilişki şekil 5.4'te gösterilmiştir.



Şekil 5.4 – YBS ile VTYS arasındaki ilişki

### 5.3. Karar Destek Sistemleri (KDS)

En genel anlamıyla KDS, yönetici konumundaki karar vericilerin karar vermelerinde yardımcı olan sistemlerdir. Diğer bir deyişle, verilmesi gereken kararlar ilgili veriyi daha iyi anlayarak, daha etkin karar seçeneklerini oluşturma, alternatifleri belirleme ve değerlendirme işlevlerinde destek sağlayan ve doğru karar verme olasılığını artıran sistemlerdir. KDS şu şekilde tanımlanabilir:

Bir **Karar Destek Sistemi**, kullanıcıya yarı-yapısal ve yapısal olmayan karar verme işlemlerinde destek sağlamak amacıyla, karar modellerine ve verilere kolay erişim sağlayan etkileşimli bir sistemdir.

Karar Destek Sistemlerine örnek olarak portföy yönetim sistemleri ve finansal planlama sistemi gibi sistemler verilebilir.

### 5.4. Ofis Otomasyon Sistemleri

Ofis otomasyonu (OO), verileri işleyenlerin, iletişim, koordinasyon faaliyetlerini düzenleyerek ofisteki verimliliklerini artırmayı amaçlayan bilgi teknolojileri uygulamasıdır. OO uygulamalarından bazıları aşağıda verilmiştir:

- Kelime işlemci, yazılı belgeleri oluşturan, görüntüleyen, biçimleyen ve yazıcıya gönderen yazılımlardır ve ilk OO uygulamasıdır.
- Elektronik posta ve sesli posta (voice mail), telefonla haberleşmeye alternatif olarak geliştirilmiş OO uygulamalarıdır.
- Faks ve bilgisayarlar kullanılarak uzun mesafeli belge kopyalamaları yapmak, bir OO uygulamasıdır.
- Elektronik takvimler, firmadaki çalışanların zamanlarını ayarlamalarını kolaylaştıran OO uygulamalarıdır.
- Son yıllarda kullanımı yaygınlaşan görüntülü ve görüntüsüz telekonferans sistemleri de bir OO uygulamasıdır ve aynı fiziksel mekanda bir araya gelmeden de toplantılar yapabilmeye imkanı vermektedir.

### 5.5. Yapay Zeka Ve Uzman Sistemler

İnsan davranışlarını taklit etme yeteneğine sahip makineler yapma isteği Yapay Zeka kavramının ortaya çıkmasına neden olmuştur. Burada esas ilgi, anlama, sonuç çıkarma, öğrenme, bilgi toplama, kendi kendini iyileştirmek için gayret sarf etme özelliğine sahip sistemler yapmak üzerinedir.

**Yapay Zeka (YZ)**, bilgisayar biliminin akıllı, yani dili kullanabilme, öğrenme, akıl yürütme, problem çözme gibi niteliklere sahip bilgisayar sistemleri tasarlamakla uğraşan koludur.

Belli başlı YZ alanları şunlardır:

- i) Bilgiye dayalı ve uzman sistemler
- ii) Doğal diller
- iii) İnsanın duyumsal yeteneklerinin taklidi
- iv) Robotik

Bu alanları kısaca inceleyelim:

### 5.5.1 Bilgiye dayalı ve uzman sistemler

Bir **Bilgi Tabanlı Sistem**, bir hastanın rahatsızlığının teşhis edilmesi gibi çözümü uzmanlık gerektiren herhangi bir problemin çözümünde IF-THEN kurallarını uygulayan bir bilgisayar tabanlı sistemdir.

Bilgi tabanlı sistemler, YBS ve KDS'de olduğu gibi gerçek bilgileri baz alırlar. Ancak bu sistemler, sezgi, yargı, vb. sezgisel bilgilerle doldurulan bilgi tabanına da ihtiyaç duyarlar.

### 5.5.2 Doğal Diller

Son kullanıcının doğal dili ile verdiği yürütme talimatlarının bilgisayar sisteminde kabul edilmesi ve ondan anlam çıkarılmasını sağlayan yazılımlardır.

Örneğin kullanıcı, bilgisayara “*Nisan ayında Quba bölgesinde yapılan toplam satış miktarı ne kadardır?*” sorusunu yöneltip sonucu bir tablo şeklinde bilgisayardan alabilir.

### 5.5.3 İnsanın duyumsal yeteneklerinin taklidi

İnsanın duyumsal yeteneklerine (görme, işitme, konuşma, hissetme vb) sahip bilgisayarlar, aynı insanlar gibi çevre ile iletişim kurma becerisine sahip olabilmektedir. Örneğin;

- Arabadan *kemerini bağla* ya da *el frenini indir* gibi sesli ikazların gelmesi. (konuşma).
- Bazı bilgisayar programlarında mikrofona söylediğiniz komutların yerine getirilmesi (işitme, konuşmayı algılama).
- Dijital kameralar yardımıyla bilgisayarların nesnelere (objelere) tanıyabilmesi (görme).

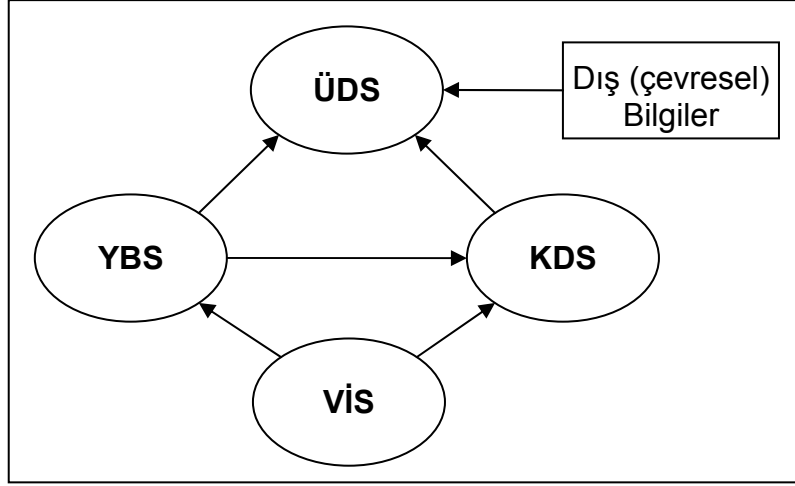
### 5.5.4 Robotik

Robotikte esas amaç, insanın fiziksel hareketlerinin taklit edilmesidir. YZ alanındaki en büyük başarılar robotlardır. Otomotiv endüstrisi gibi çeşitli endüstrilerde pek çok işlemlerde (boyama, vida sıkma, taşıma, kusurlu parça tanıma vb) robotlardan insan gücü yerine faydalanılmaktadır.

## 5.6. Üst Yönetim Destek Sistemleri

Üst düzey yöneticiler, karar vermek için Üst Yönetim Destek Sistemlerini (ÜDS) kullanırlar. ÜDS, yeni vergi yasaları veya rakip firmalar gibi dış olaylar hakkındaki verileri birleştirmeyi ve bunları YBS ve KDS'den elde edilmiş özet bilgilerle birlikte faydalı hale getirerek üst düzey yöneticilere sunmayı amaçlar.

ÜDS'nin diğer bilgi sistemleriyle ilişkisi Şekil 5.5'te gösterilmiştir.



Şekil 5.5 – ÜDS'nin diğer sistemlerle ilişkisi

## 6. BİLGİ SİSTEMLERİNİN GELİŞTİRİLMESİNDE KULLANILAN ARAÇLAR

Bilgi sistemlerinin analiz ve tasarımında kullanılan araçlardan en önemlileri bu bölümde tanıtılacaktır.

### 6.1. Akış Şemaları

Daha önceki bölümlerde (Bölüm 2.2.6) akış şemaları hakkında bilgi verilmiş ve kullanılan semboller gösterilmişti. Akış şemaları genelde iki tiptir:

- Sistem akış şeması (süreç akış şeması)
- Program akış şeması

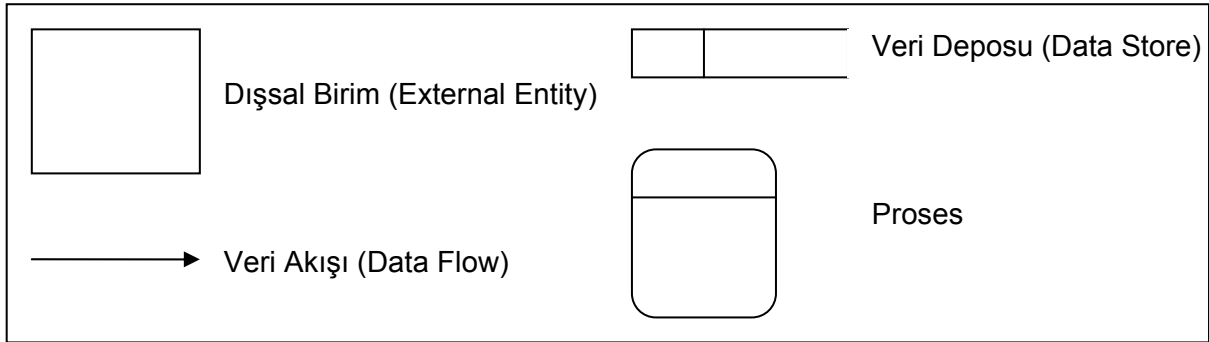
Bu akış şemalarından birincisi daha önce adı geçen bölümde anlatılmıştı. Ek1'de her iki tip akış şeması için de birer örnek verilmiştir. Kullanılan sembollerin anlamları için Bölüm 2.2.6'ya bakınız.

Sistem akış ve program akış şemalarına ilişkin olarak bir bordro sistemini gösteren örnekler şekil 6.2'de verilmiştir.

### 6.2. Veri Akış Diyagramları (VAD)

Veri Akış Diyagramları, sadece 4 adet sembol kullanarak sistemdeki veri akışını grafiksel olarak izah etmeye yarayan çok kullanışlı bir araçtır. Bilgi sistemi tasarımcıları tarafından sıklıkla kullanılan bir araçtır ve sistem ne kadar karmaşık olursa olsun bu diyagramlar sistemi tarif etmek için yeterlidir.

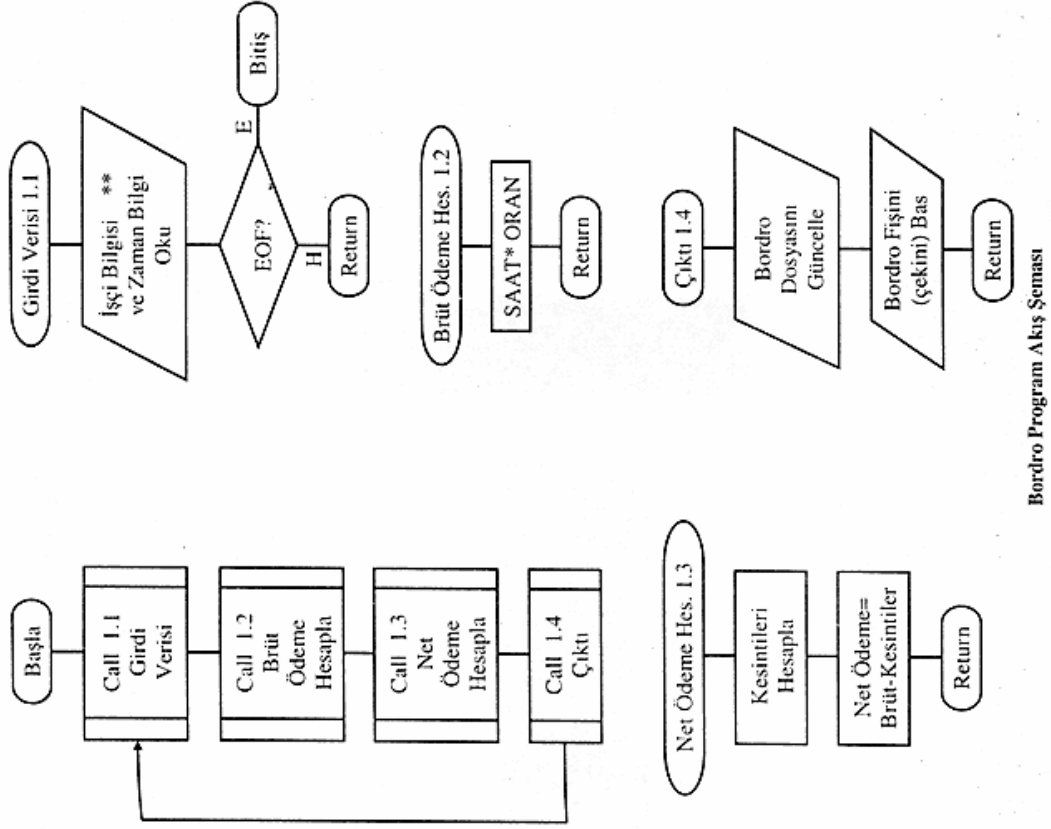
Literatürde VAD'lar için kullanılan iki standart sembol kümesi bulunmaktadır. Her iki kümede de dörder adet sembol bulunmakta ve semboller farklı olsa da aynı anlamları ifade etmektedir. Bu derste kullanılacak olan semboller Şekil 6.1'de gösterilmiştir.



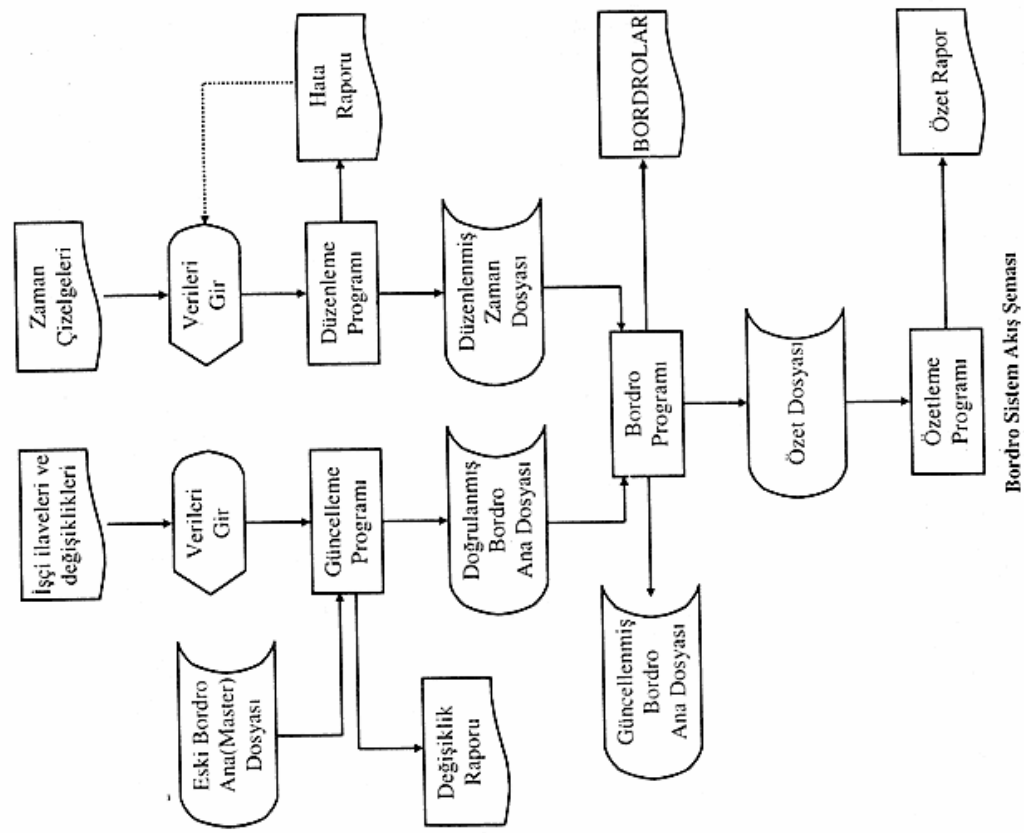
Şekil 6.1 – VAD Sembolleri

Bu sembollerin anlamlarını ve kullanım şekillerini kısaca açıklayalım:

- i) *Veri Akışı*: Bir veri akışı, bir sistemde bir yerden başka bir yere hareket eden veriyi temsil eder. Yani veri akışı hareket halindeki veridir. Veri akışı bir ok ile gösterilir ve bu ok üzerinde de o akışın içeriği yazılır. Bu içerik, tek bir veri olabileceği gibi (Kayıt No gibi) kompozit bir veri de olabilir (satış raporları gibi).
- ii) *Proses*: Prosesler, yapılan bir fonksiyonu ya da aktiviteyi tanımlar. Proseslere genelde bir isim ve numara verilir. Bu numaralar proses sırasını gösteren numaralar değildir. Proses ismi olarak da emir cümleleri kullanmak uygun olacaktır (Brüt maaşı hesapla gibi).



\*\*\* : Düzenlenmiş zaman dosyasından zaman verileri ve doğrulanmış bordro ana dosyasından işçi verileridir.

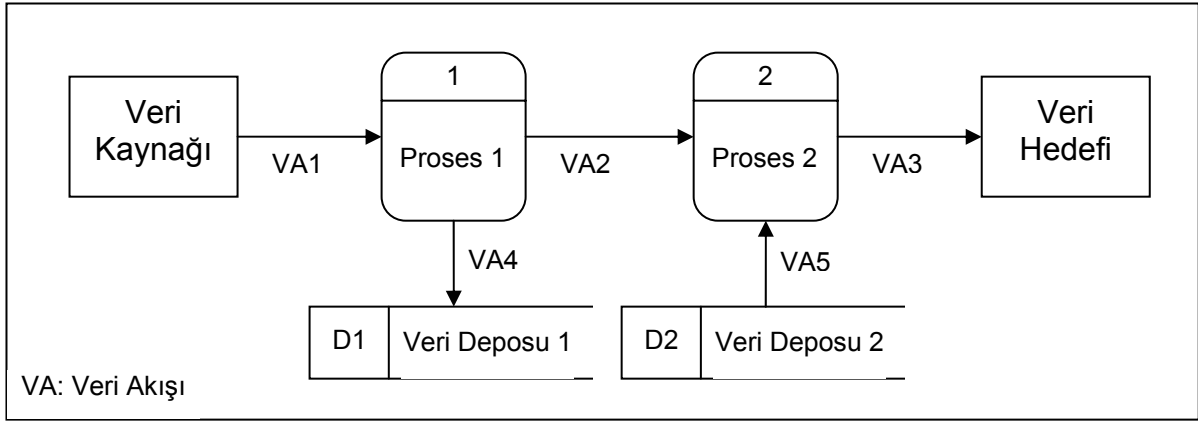


Şekil 6.2 – Sistem Akış ve Program Akış Şeması Örnekleri

iii) *Dışsal Birimler*: Bu birimler, veri/bilgi kaynağı ya da verinin/bilginin gideceği yerdir. Sisteme veri sağlayan ya da sistemden veri alan birisi bu tanıma örnektir. Birimin adı sembolün içine tekil olarak yazılır ve sembolün sol üst köşesinde de bu birimi tanımlayan bir harf bulunabilir. Veri akış çizgilerinin kesişmesini önlemek için aynı birim aynı diyagramda birden çok defa kullanılabilir. Aynı birim aynı diyagramda birden çok defa kullanılıyorsa sembolün sağ alt köşesine bir diagonal çizilir.

iv) *Veri Deposu*: Analiz esnasında, verilerin depolanmasına ihtiyaç duyulan yerler olur. Bu yerler veri deposu olarak isimlendirilir. Veri deposu, bir raf, dosya kabini ya da bilgisayar dosyası olabilir. Her bir veri deposu D ile tanımlanır ve referans olması amacıyla D'nin yanına bir rakam verilir. Her bir veri deposu için ayrıca bir de isim verilir. Dışsal birimde olduğu gibi aynı veri deposu aynı diyagramda birden çok kullanılırsa sembolün sol tarafına dikey bir çizgi çekilir.

Genel bir veri akış diyagramı şekil 6.2'de verilmiştir.



Şekil 6.3 – Genel Bir VAD

Çoğu organizasyonların ya da sistemlerin çalışması oldukça karmaşıktır ve onları aynı düzlemde tek bir veri akış diyagramı ile ifade etmek oldukça zordur. Bu nedenle, diyagramın alt seviyeler bölünmesi gerekir.

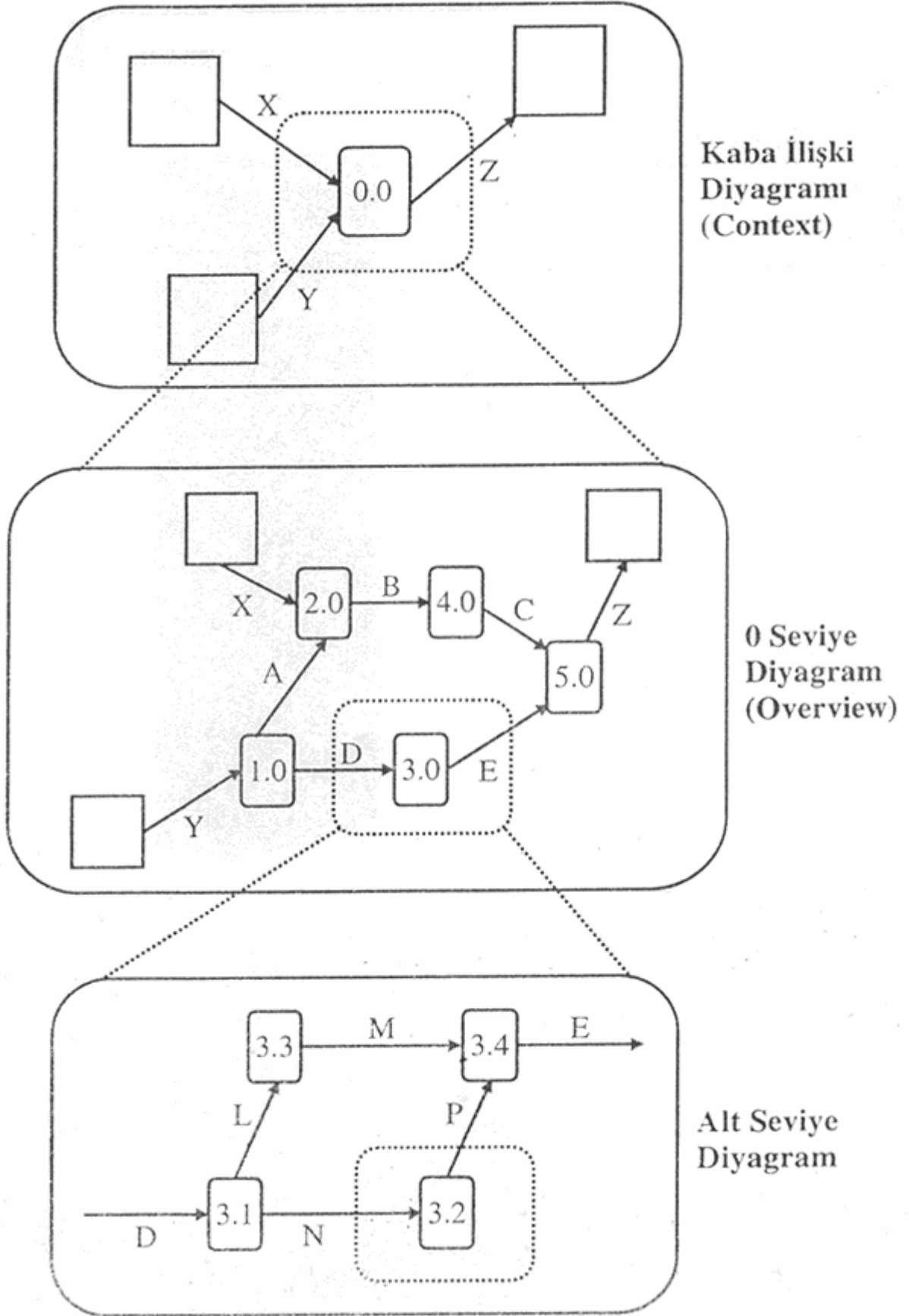
VAD'ın en yüksek seviyeli hali, Kaba İlişki (Context) diyagramı olarak adlandırılır. Kaba İlişki Diyagramı (KİD) çizilirken sadece tek bir proses sembolü kullanılır. Daha sonra yapılacak seviyelendirmeye uygun şekilde de numaralandırma yapılır. Örneğin KİD içindeki proses için 0 numarası verilir daha sonraki seviyeler için 1.0, 2.0 gibi numaralandırmaya devam edilebilir. En üst seviye diyagram yani KİD, 0-seviye diyagramıdır. VAD'da seviyelendirmenin nasıl yapıldığı, şekil 6.4'te incelenebilir.

### 6.2.1 VAD Çizim Kuralları

Bir VAD çizilirken uyulması gereken kurallar aşağıda listelenmiş ve bu kuralların bazılarına ilişkin şekilsel gösterim Şekil 6.5'te sunulmuştur.

#### PROSES

- Hiçbir proses sadece çıktılara sahip olamaz. Girdi olmadan dönüşüm yapılamaz. Eğer böyle bir birim var ise kaynak birim olmalıdır.
- Hiçbir proses sadece girdilere sahip olamaz. Böyle bir birim varsa bu hedef birim olmak zorundadır.
- Bir proses fiil sözcük öbekli isme sahip olmalıdır.



Şekil 6.4 – VAD'da Seviyelendirme

## VERİ DEPOSU

- D. Veri, bir veri deposundan diğerine doğrudan taşınmaz. Veri bir prosesle taşınmalıdır.
- E. Veri, doğrudan bir dışsal kaynaktan bir veri deposuna taşınmaz. Dışsal birimden veriyi alan ve veri deposuna yerleştiren bir prosesle taşınmalıdır.
- F. Veri, bir veri deposundan doğrudan bir dışsal birime taşınmaz. Veri bir prosesle taşınmalıdır.
- G. Bir veri deposu isim sözcük öbekli bir isme sahip olmalıdır.

## DIŞSAL BİRİM

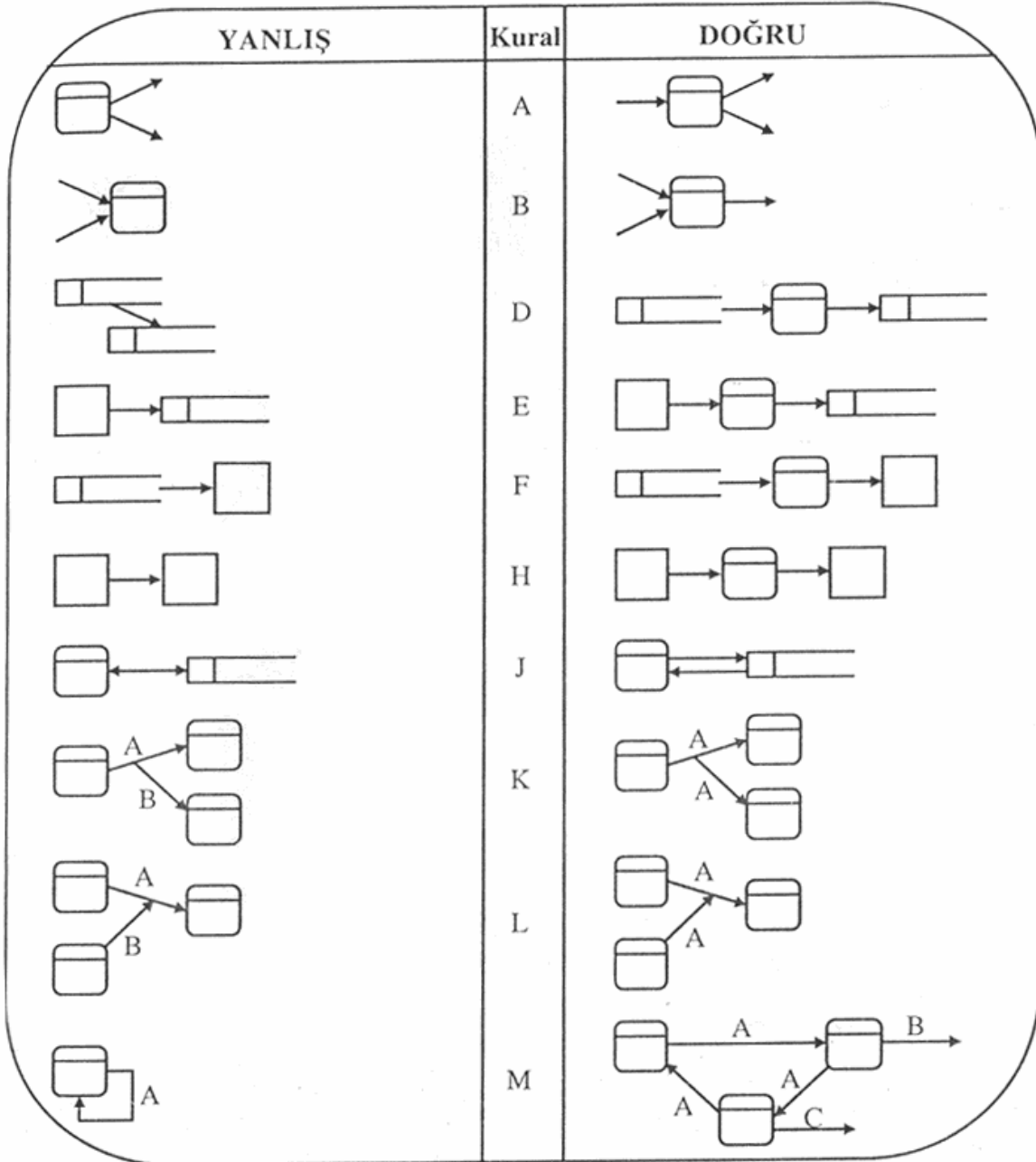
- H. Veri doğrudan bir dışsal birimden diğerine taşınmaz. Veriler sistemle ilgiliyse, bir prosesle taşınmak zorundadır. Değilse zaten VAD içinde gösterilmez.
- I. Bir dışsal birim isim sözcük öbekli bir isme sahip olmalıdır.

## VERİ AKIŞI

- J. bir veri akışı, semboller arasında tek bir akış yönüne sahip olmalıdır. Bir proses ve veri deposu arasında, veri deposundan okuma ve proseste güncellenmenin gösterilmesi için her iki yönlü akış olabilir, ancak bunların iki ayrı ok şeklinde gösterilmesi gerekir.
- K. Çatallı bir veri akışının anlamı, aynı verinin ortak bir lokasyondan iki ya da daha fazla farklı prosese, veri deposuna ya da dışsal birime gitmesi demektir.
- L. Veri akışlarının birleşmesinin anlamı, aynı verinin herhangi iki ya da daha fazla farklı prosesten, veri deposundan ya da dışsal birimden, ortak lokasyona gelmesidir.
- M. Bir veri akışı, doğrudan aynı prosese geri dönemez. Veri akışını alıp, başka veri akışlarını üreten ve başladığı prosese orijinal veri akışını getiren en az bir prosesin olması gerekir.
- N. Bir veri akışının veri deposuna gitmesinin anlamı, güncellemedir.
- O. Bir veri deposundan veri akışının çıkmasının anlamı, getirme ya da kullanmadır.
- P. Bir veri akışı, isim sözcük öbekli bir etikete sahip olmalıdır. Birden fazla veri akış isim sözcük öbeği, tek bir ok üzerinde görülebilir.

Bunların dışında;

- prosesin girdileri, prosesin çıktılarından farklı olmalıdır.
- VAD üzerindeki birimler tek isme sahip olmalıdır. Örneğin aynı isimli iki proses sahip olmanın anlamı yoktur. VAD çizilirken karışıklığın önlenmesi için veri deposu ve dışsal birimler tekrar edebilir. Eğer iki ok aynı veri akış ismine sahipse, bu akışların tamamen aynı olmaları gerekir. Bir akış ismi her zaman aynı şeyi ifade etmelidir.



Şekil 6.5 - Çizimindeki Yanlışlıklar ve Doğru Çizim Yolları

### 6.2.2 Mantıksal-Fiziksel VAD

Bir bilgi sistemi için VAD'ın çizilme amacı, sistem boyunca veri akışlarının modellenmesidir. VAD çizilirken ilk etapta sistemi anlamamıza yardımcı olan fiziksel objelerle karşılaşılır. Eğer çizilen VAD içinde fiziksel objeler var ise bu *Fiziksel VAD* olarak adlandırılır. *Mantıksal VAD* ise fiziksel objelerin olabildiğince ayıklandığı diyagramlardır.

Fiziksel VAD, mantıksal VAD'ın çizilmesinde bir araç olarak kullanılabilir. Bu açıdan ilk başta fiziksel VAD çizmek faydalı bir yaklaşım olarak kabul edilebilir. Çeşitli VAD örnekleri Ek1'de verilmiştir:

Çeşitli VAD örnekleri:

*Ek1 – Çeşitli VAD Örnekleri / YBS Kitabı (Sayfa 85,95)*

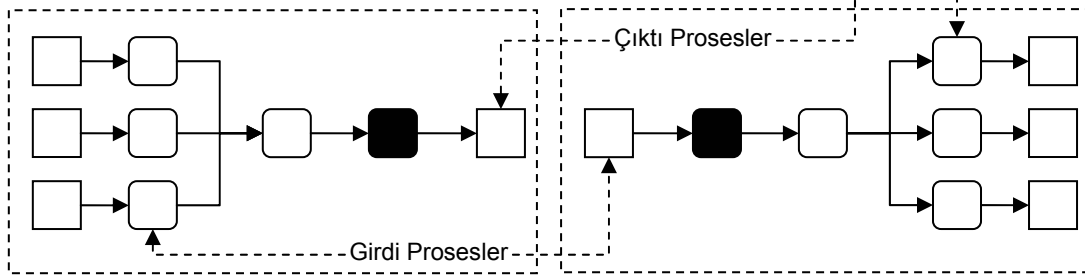
### 6.3. Yapısal Şemalar

Yapısal şemalar (YŞ), hiyerarşik bir yapı içerisinde program modüllerinin ve bu modüllerin diğerleriyle ilişkisinin grafiksel gösterimidir. Bir yapısal şemanın en üstünde (root), tek bir yönetici modül bulunur. Bir sonraki seviyede ise, modül çağrılmalarını koordine eden diğer modüller bulunmaktadır. En alt seviyedeki modüller, herhangi bir başka modül çağırılmazlar, sadece belirli görevleri yaparlar.

VAD sistemin NE yaptığını gösteren bir analiz aracıdır, YŞ ise sistemin bilgisayar programcılarınca NASIL yürütüleceğini gösteren bir tasarım aracıdır.

Yapısal şemalarda iki teknik kullanılır: [1] Dönüştürme (Transform) Analizi ve [2] İşleme (Transaction) Analizi. Bilgi sistemleri genelde ya dönüştürme merkezli ya da işleme merkezlidir. Yapısal şemalar çizilirken önce sistemin dönüştürme ya da işleme merkezi tespit edilmelidir.

İşleme merkezli sistemde, veri sistemin merkez modülüne (işleme merkezi) gelir, değerlendirilir ve oradan uygun lokasyonlara dağıtılır. Dönüştürme merkezli sistem yeni değerler üreten bir merkezi dönüştürme modülüne sahip olmakla beraber, bu sistemlerin girdileri fazla çıktıları azdır. Bu iki tip sistem şekil 6.3'de gösterilmiştir.



Şekil 6.6 – Dönüştürme ve İşleme Merkezli Sistemler

Çeşitli Yapısal Şema (YŞ) örnekleri:

*Ek2 – Çeşitli Yapısal Şema Örnekleri / YBS Kitabı (Sayfa 98,105)*

### 6.4. Yapısal Dil

Çoğu durumlarda, bilgi sistemi tasarımı için kullanılan akış şemaları, karar tabloları ve HIPO gibi araçlardan gerçek programlara geçmek oldukça zor olabilir. YD, VAD'da bulunan proseslerdeki dönüşüm işlemlerinin nasıl yapılacağını tarif etmek için kullanılır. YD, bir nevi normal konuşma dilini kullanarak bilgisayar programları yazmaya benzer. YD, Sahte Kod (SK) (Pseudocode) olarak da bilinir. Bu iki kavram arasında temelde bir fark olmamakla beraber YD'nin konuşma diline, SK'nın ise programlama diline daha yakın olduğu düşünülebilir.

Aşağıda bir YD örneği verilmiştir:

**Örnek:** Firmada Ayda brüt 250 dolardan fazla kazananların listesi

- 1- PRINT Rapor Başlığı
- 2- READ Her bir Personel Verisi
- 3- Brüt Ödemeyi Hesapla
- 4- Brüt Ödeme 250 Dolar'dan Fazla mı?
  - a. Evet ise, PRINT Numara, Oran, Brüt Ödeme
  - b. Hayır ise, Hiçbir şey Yazma
- 5- Tüm personel için 2-4 adımları tekrarla

Yapısal dil normal olarak READ, WRITE, SORT, MOVE, MERGE, SUBTRACT, MULTIPLY, DIVIDE, DO, FINE gibi İngilizce fiilleri kullanır. Ayrıca değişkenler için MUSTERI\_ADI, MUSTERI\_ADRESI gibi etiketler kullanılır. Sıfat ve zarf kullanılmaz ve kullanılan isimler *veri sözlüğünden (ileride anlatılacak)* alınırlar.

Bilgisayar programlarındaki tipik 3 prosesin gösterilmesinde YD kullanılabilir. Bunlar: [1] Sıra (sequence), [2] şartlı ifadeler (conditional statements), [3] tekrar (repetition).

Sıra, programlamada özel bir yapı gerektirmeyen sıralı işlemleri ifade eder;

İşlem 1...  
İşlem 2...  
İşlem 3...

gibi.

Şartlı ifadeler ise herhangi bir mantıksal ya da matematiksel ifadenin farklı durumlarında ne yapılması gerektiğini gösterir. Örneğin;

```
IF şart A doğruysa
    İşlem A'yı yap
ELSE İşlem B'yi yap
END IF
```

```
IF Stoktaki_Miktar, Min_Stok_Miktarı ndan küçükse
    THEN Yeni sipariş üret
ELSE Bir şey yapma
END IF
```

Şartlı ifadelerin bir diğer şekli de programın izleyebileceği birçok farklı yol olduğu zamanlarda kullanılır. Bunun için CASE ifadesi kullanılır, şöyle ki;

```
SELECT CASE
CASE 1 (Şart 1)
    Şart 1 için yapılacak işlemler
CASE 2 (Şart 2)
    Şart 2 için yapılacak işlemler
.
.
CASE n (Şart n)
    Şart n için yapılacak işlemler
END CASE
```

Bir örnek verecek olursak;

```
READ Stoktaki_Miktar
    SELECT CASE Stoktaki_Miktar
        CASE 1 (Stoktaki_Miktar, Min_Sipariş_Miktarı ndan büyük ise)
            Hiçbirşey yapma
```

CASE 2 (Stoktaki\_Miktar, Min\_Sipariş\_Miktari na eşit ise)

İlgili kişiye e-posta gönder, durumu bildir

CASE 3 (Stoktaki\_Miktar, Min\_Sipariş\_Miktari ndan küçük ise)

Yeni Sipariş Üret

CASE 4 (Stok Yoksa)

Acil olarak yeniden sipariş yöntemi çalışması başlat

END CASE

Tekrarlı işlemlerde ise DO-UNTIL ya da DO-WHILE benzeri ifadelerle döngüler oluşturulur. Örneğin;

DO

READ Stok Kayıtları

IF Stoktaki\_Miktar, Min\_Sipariş\_Miktari ndan küçük ise

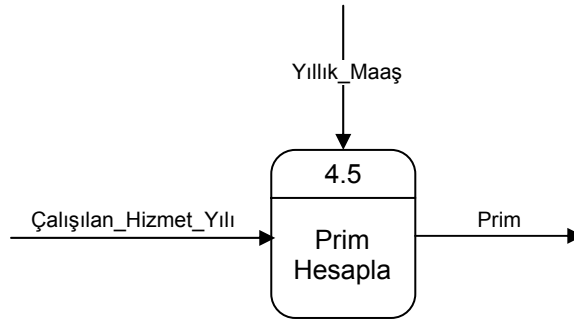
THEN Yeni sipariş üret

ELSE Hiçbir şey yapma

END IF

UNTIL EOF (Dosyada Kayıt Kalmayana Kadar)

Aşağıda bir VAD içerisinde bulunan bir proses için hazırlanmış bir YD örneği verilmiştir:



IF Çalışılan\_Hizmet\_Yılı 25 ya da daha büyükse THEN

Yıllık\_Maaş la 0.05 i Çarparak Prim i Hesapla

ELSE (Çalışılan\_Hizmet\_Yılı 25 den azdır)

Yıllık Maaş la 0.025 i Çarparak Prim i Hesapla

END IF

## 6.5. Karar Tabloları

Eğer proses mantığında birçok şartlı durum söz konusuysa ve her bir durum farklı işlemlerin yapılmasını gerektiriyorsa, o zaman karmaşık mantığın anlaşılması, YD ile oldukça zordur. Yapılan araştırmalar, çok sayıda iç içe yuvalanmış IF ifadelerinin yorumlanmasında, insanları bunları birbirine karıştırdıklarını göstermiştir. Proseslerin böyle karmaşık olduğu durumlarda Karar Tabloları durumu YD'den daha iyi ifade edebilir.

Karar tablosu, sistemin mantığını adım adım yerine tablo biçiminde gösteren bir araçtır. Akış şemalarının alternatifi olabileceği gibi birlikte de kullanılabilirler.

Karar tabloları 4 ana bölümden oluşur:

- 1) Tablonun sol üst bölümüne mümkün olan tüm şartlar yazılır.
- 2) Sol alt bölümde, şartların kombinasyonları sonucu yapılan tüm mümkün faaliyetler listelenir.
- 3) Kurallar, sağ üst bölümde oluşturulur. Şartın durumuna bağlı olarak şartın karşılığındaki olasılıkları ifade eden gösterimler kullanılır. Örneğin; Evet, Hayır'ı temsil için E ve H harfleri kullanılır.
- 4) Sağ alt bölümde ise, verilen bir kural için geçerli olan faaliyetlerin gösterilmesini sağlayan ve onay anlamında olan "X" işaretleri yerleştirilir.

Karar tablosundaki geçerli teorik kural sayısı  $2^n$ 'dir. Burada n şart sayısıdır. Her şartın karşılığında sadece 2 olasılık olduğu durumlarda geçerli olan bu formüle göre eğer 3 şartımız varsa toplam kural sayımız  $2^3 = 8$  olur.

Tablo 6.1'de bir karar tablosu örneği verilmiştir. Bu teorik olarak tüm kuralları içeren bir karar tablosudur. Tablo 6.2'de ise aynı sisteme ait sadece pratikte mümkün olan kuralları içeren bir karar tablosu verilmiştir. Bu indirgenmiş karar tablosudur.

İndirim Hesapla	KURALLAR																															
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
<b>ŞARTLAR</b>																																
Satın alma<100\$	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	
Özel Teklif	E	E	E	E	E	E	E	E	H	H	H	H	H	H	H	E	E	E	E	E	E	E	E	H	H	H	H	H	H	H	H	
İndirim<2\$	E	E	E	E	H	H	H	H	E	E	E	E	H	H	H	H	E	E	E	E	H	H	H	H	E	E	E	E	H	H	H	
İndirim Sonrası>45\$	E	E	H	H	E	E	H	H	E	E	H	H	E	E	H	H	E	E	H	H	E	E	H	H	E	E	H	H	E	E	H	
7 Gün içinde ödeme	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	
<b>FAALİYETLER</b>																																
2\$ indirim	X	X	X	X																												
%5 indirim									X	X	X	X	X	X	X	X																
%7.5 indirim					X	X	X	X																								
%8 indirim																	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
Ekstra %1 indirim				X				X			X					X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

Tablo 6.1 – İndirim Hesaplama Prosesi İçin Karar Tablosu

İndirim Hesapla	KURALLAR								
ŞARTLAR	1	2	3	4	5	6	7	8	9
Satın alma<100\$	E	E	E	E	E	E	E	H	H
Özel Teklif	E	E	E	E	H	H	H	-	-
İndirim<2\$	E	H	H	H	-	-	-	-	-
İndirim Sonrası>45\$	-	E	-	H	-	E	H	-	-
7 Gün içinde ödeme	-	E	H	-	H	E	E	E	H
FAALİYETLER									
2\$ indirim	X								
%5 indirim					X	X	X		
%7.5 indirim		X	X	X					
%8 indirim								X	X
Ekstra %1 indirim		X				X	X		

Tablo 6.2 – İndirgenmiş Karar Tablosu

Karar tablosunda bazı şartların birden fazla karşılığı olabilir örneğin;

	KURALLAR					
ŞARTLAR	1	2	3	4	5	6
Çalışan Tipi	M	S	S	A	S	A
Çalışılan saatler	<40	<40	40	40	>40	>40
FAALİYETLER						
Taban ücreti öde	X		X		X	
Saatlik ücret hesapla		X		X		X
Fazla mesai ücreti hesapla					X	X
Devamsızlık raporu üret		X				

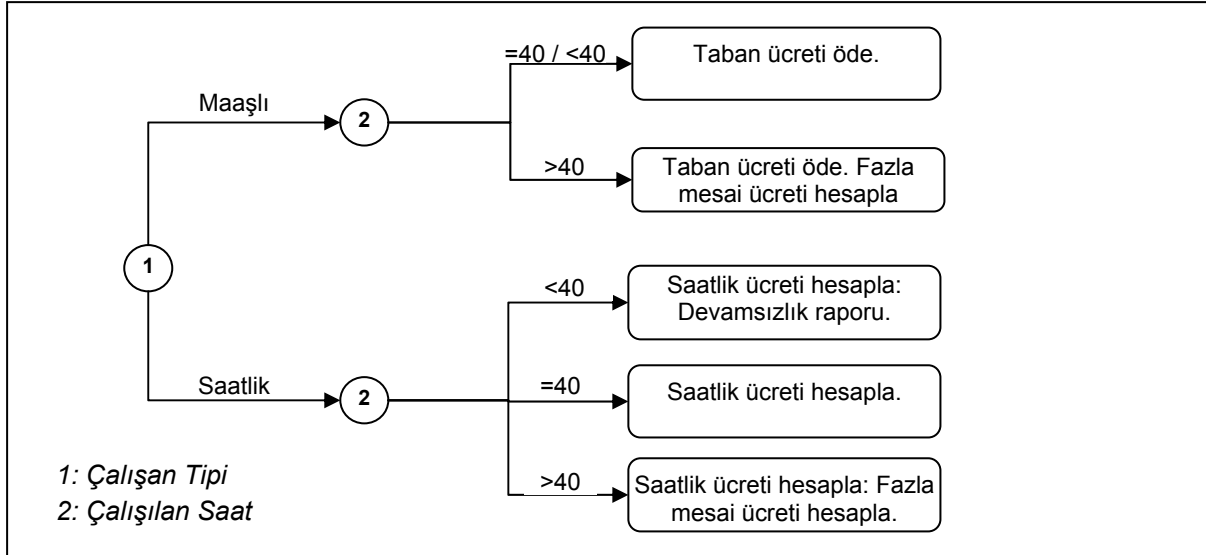
M: Maaşlı S: Saat ücretli

Tablo 6.3 – Şartların ikiden fazla olasılık taşıdığı durum için örnek (Karar Tablosu)

## 6.6. Karar Ağaçları

Karar ağaçları da karar tabloları gibi çok sayıda şartlı ifadelerle sahip proseslerin tanımlanmasında kullanılabilir. Karar ağaçları, karar tabloları için bir alternatif araçtır. Programların daha etkin yazımında karar tablolarından ziyade karar ağaçlarının daha faydalı olduğunu gösteren araştırmalar vardır. Bunun nedeni grafiksel gösterim ve adımların akışının görülebilmesidir.

Bu karar ağaçları yönetim bilminde kullanılan karar ağaçlarından farklı olarak olasılık içermezler. Çünkü sistem analizindeki karar ağaçlarının amacı karar prosesindeki şartlı durumları ve faaliyetleri tanımlamak ve organize etmektir. Bir karar ağacı örneği şekil 6.4'te verilmiştir.



Şekil 6.7 – Karar ağacı örneği

## 6.7. HIPO

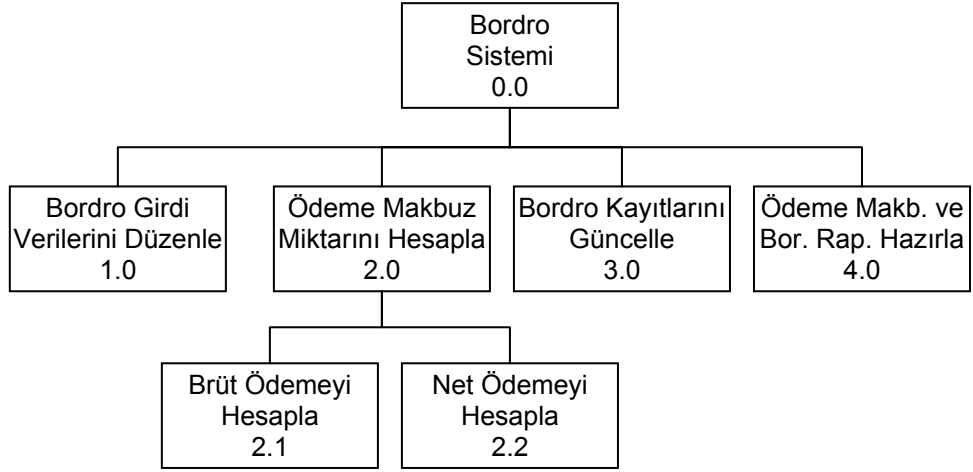
Bilgi sistemi geliştirme araçlarından bir diğeri de IBM tarafından büyük ve karmaşık çalışma sistemleri için geliştirilmiş olan HIPO (Hierarchy Plus Input-Processing-Output) tekniğidir. “Nasıl” dan ziyade “Ne” yapılacağı üzerinde yoğunlaştığı için akış şemalarından farklıdır. HIPO'nun 3 temel amacı vardır:

- 1) Sistem fonksiyonlarının parçalara ayrılmış hiyerarşik yapısını göstermek
- 2) Sistem fonksiyonlarının ayrıntılarını herhangi bir programlama diline bağlı kalmaksızın göstermek.
- 3) Sistem fonksiyonları düzeyinde, girdiler ve çıktıları görsel olarak tarif etmek.

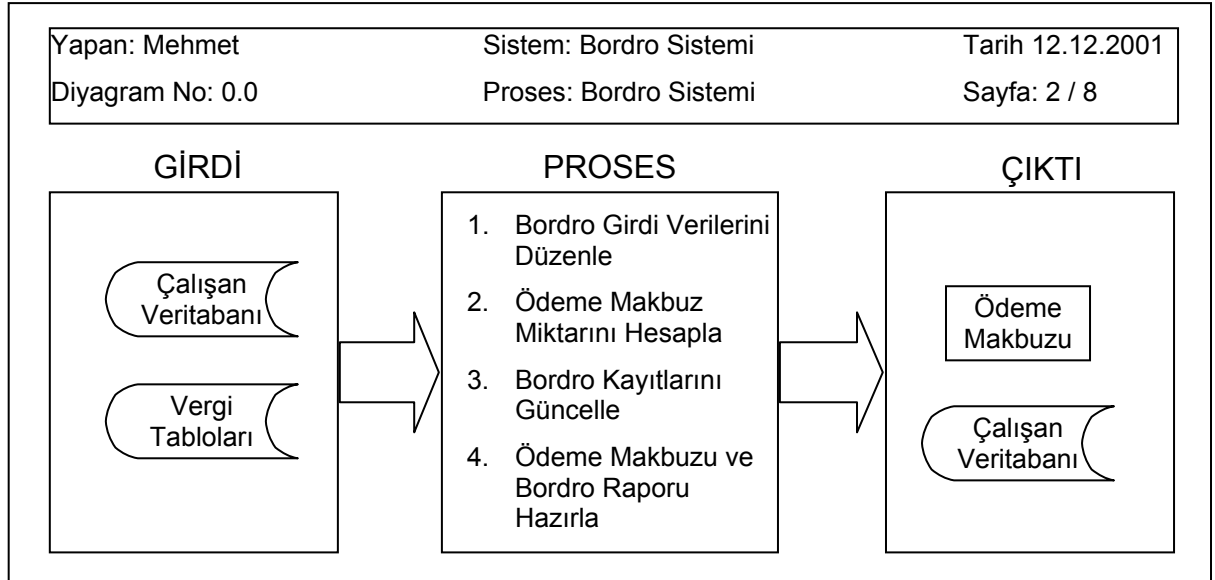
HIPO, iki ayrı diyagramdan oluşur:

- 1) Görsel İçerik Tablosu: Hiyerarşi diyagramı olarak da bilinir. İngilizce kısaca VTOC (Visual Table Of Contents) olarak ifade edilir. Sistemi yukarıdan aşağıya hiyerarşik bir yapıda modüller olarak ifade eden bir şemadır.
- 2) HIPO Özet Diyagramı: İngilizce HIPO Overview Diagram olarak bilinir. VTOC'teki her bir kutu (modül) için girdi, çıktı ve ana prosesleri gösterir.

Şekil 6.5 ve 6.6'da bir bordro sistemine ait VTOC ve HIPO özet diyagramı örneği verilmiştir.



Şekil 6.8 – Bordro Sistemi için HIPO Hiyerarşi Diyagramı



Şekil 6.9 – HIPO Özet Diyagramı

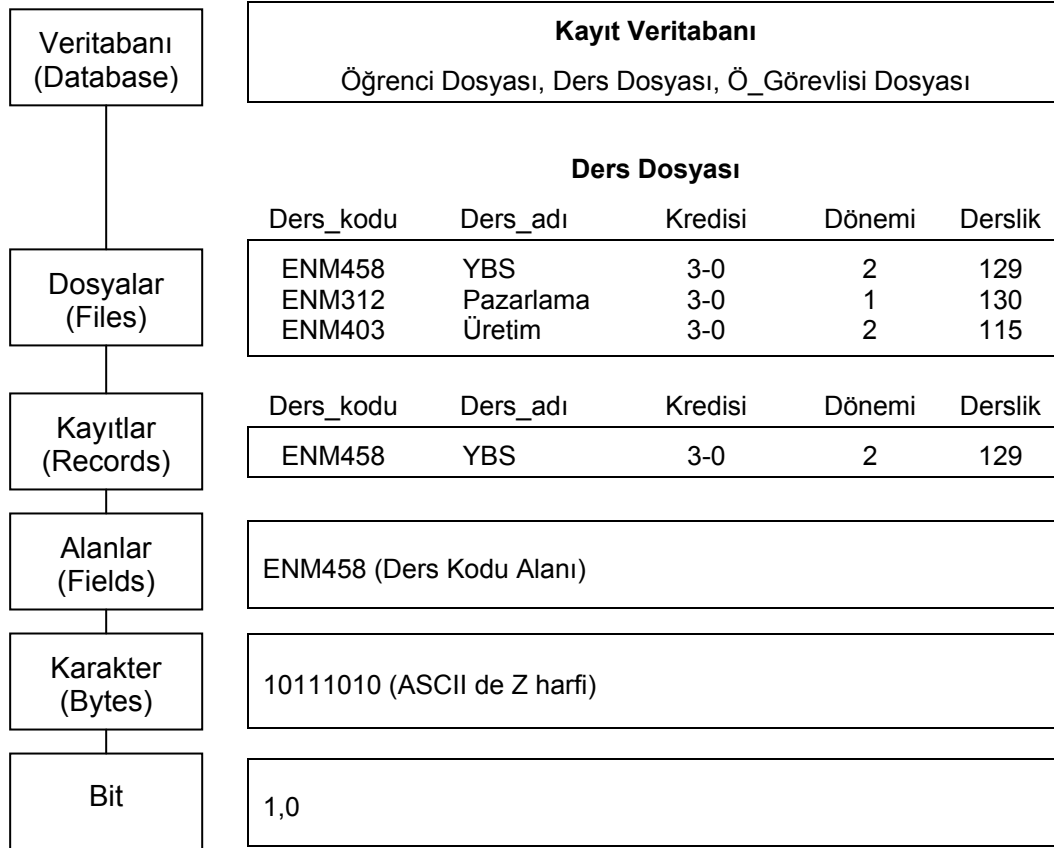
## 7. VERİTABANI TASARIMI ve YÖNETİMİ

### 7.1. Veri Yönetimi

Veri yönetimi, iç ve dış kaynaklardan elde edilen verilerin gerektiğinde bilgiye dönüştürülebilmesi için saklanması, çağırılması ve üzerinde birtakım işlemler yapılmasını içerir. Veri yönetimiyle ilgili bazı kavramlar aşağıda açıklanmıştır. Bu kavramların ardından, veri yönetimi yaklaşımları hakkında bilgi verilmiştir.

#### 7.1.1 Veri Hiyerarşisi

Verilerin etkin bir şekilde faydalı bilgilere dönüştürülebilmesi için önce anlamlı bir şekilde organize olması gereklidir. Veriler genellikle şekil 7.1’de gösterilen hiyerarşi ile organize edilirler.



Şekil 7.1 – Veri Hiyerarşisi

#### 7.1.2 Birimler (Entity), Özellikler (Attributes) ve Anahtarlar

Bir birim, varolan ve ayırt edilen herhangi bir şey olarak tanımlanabilir. Bir kayıt, bir veri birimini tanımlar. Bu bir kişi, bir yer ya da herhangi bir malzeme kalemi olabilir. Siparişler, çalışanlar, stoklar, müşteriler veri birimi için örnek olarak tanımlanabilirler.

Veri birimlerinin özellikleri vardır. Örneğin, bir ders dosyasındaki Ders\_Kodu, Ders Adı, Kredisi, Dönemi ve Derslik, ders veri biriminin özellikleridir.

Tablo 7.1’de her bir sütun bir alanı yani bir özelliği, her bir satır da bir kaydı temsil etmektedir.

Kayıtlarda iki kaydı birbirinden ayıran bir alanın olması gerekir. Bu özellik kayıtlara erişimi kolaylaştırmaktadır. Her bir kaydı tek (benzersiz, unique) olarak tanımlamak

için kullanılan, kayıttaki bir alana ya da alanlara birincil anahtar (primary key) denilmektedir. Örneğin Tablo 7.1'deki Öğrenci No, her bir öğrenci için benzeri olmayan bir sayıdır. Bu sayı öğrenciyi tarif etme imkanı tanıyan birincil anahtardır. Aynı isimli ya da soyadlı öğrenciler olabilse de aynı numaraya sahip öğrencilerin olması olanaksızdır. Buradan yola çıkarak her bir veri biriminin birincil anahtarı olması gerektiği söylenebilir. Ancak bir veri birimi, hiçbir birincil anahtar olabilecek durumda olmayan özelliklerden meydana gelmişse bir kaydı diğerinden ayırmak zorlaşacaktır. Bu gibi durumlarda çoğunlukla bir sıra numarası birincil anahtar olarak veri birimine eklenmektedir.

Öğrenci No	Soyadı	Adı	Diğer Özellikler...
2011203	Kadir	Muharrem	*
2011223	Zekeriyagil	Fatma	*
2011234	Selamioğlu	İlyas	*
2011324	Kadir	Fatma	*

Tablo 7.1 – Öğrenci Dosyası (Veri Birimi)

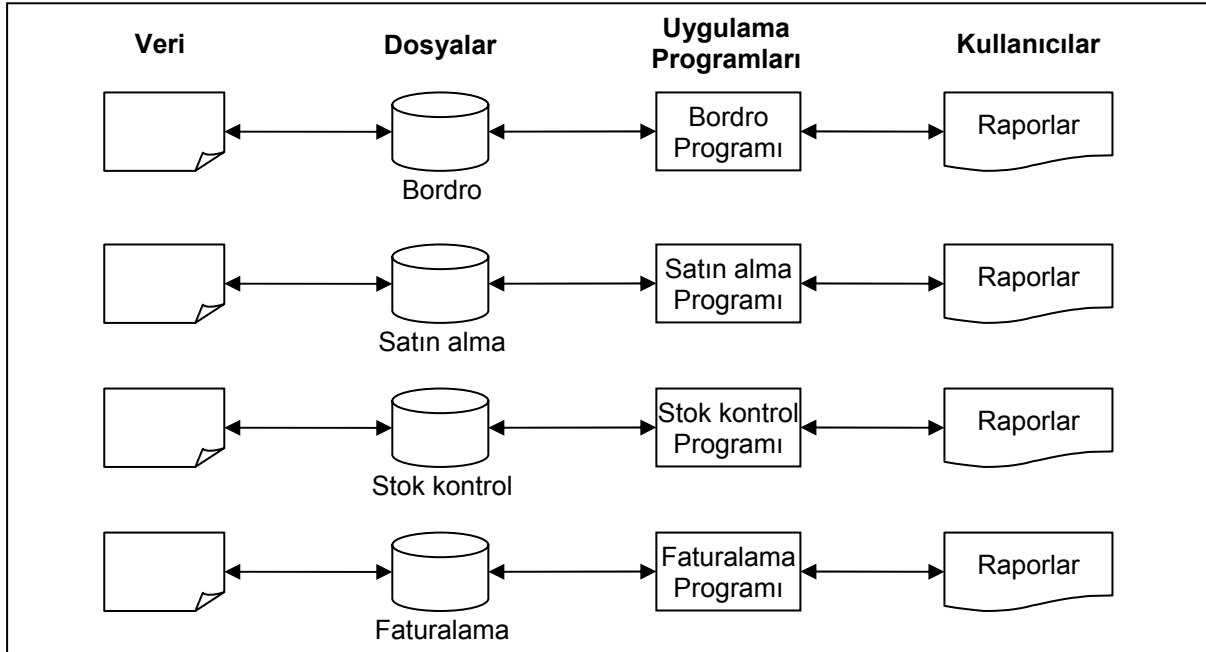
### 7.1.3 Veri yönetimi yaklaşımları

Veritabanı Yönetim Sistemleri (VTYS), bir organizasyonun veriyi etkin bir şekilde idare etmesine izin veren ve uygulama programlarınınca, saklanan veriye erişilmesine imkan sağlayan yazılımlardır (Oracle, Sybase, IBM DB2 gibi). Verinin yönetilmesinde iki yaklaşımdan söz edilebilir.

- i) Geleneksel Yaklaşım                      ii) Veritabanı (VT) Yaklaşımı

#### 7.1.3.1 Geleneksel yaklaşım

Bu veri yönetimi yaklaşımı, dosya kökenlidir. Bu yaklaşımda her bir uygulama kendi dosyalarıyla yürütülmektedir. Yani her bir uygulama problemi için ayrı veri dosyaları yaratılmakta ve saklanmaktadır. Şekil 7.2, bu yapıyı göstermektedir.



Şekil 7.2 – Geleneksel Veri Yönetimi Yaklaşımı

Geleneksel yaklaşımın bazı dezavantajları vardır. Örneğin, *veri tekrarlılığı*, yani gereksiz verilerin saklanmasıdır. Bu yaklaşımda sık sık aynı veri iki ya da daha fazla dosyada saklanabilmektedir. Tablo 4.2 de bununla ilgili bir örnek verilmiştir. Bu örnekte sosyal güvenlik numarası, çalışanın adı ve bölümü verileri her iki dosyada da tutulmaktadır.

*Dosyalar arası kopukluk* bu yaklaşımın bir diğer dezavantajıdır. Örneğin yönetim herhangi iki ya da daha fazla dosyadan bazı verilerin ilişkilendirilmesini gerektiren bir rapor istediğinde bu istek karşılıksız kalacaktır. Çünkü bu yaklaşım dosyaları ilişkilendirecek bir mekanizmaya sahip değildir.

*Program/veri bağımlılığı* da bu yaklaşımın bir dezavantajıdır, şöyle ki, dosyaları güncellemek ve idame ettirmek için gerekli spesifik programlarla, dosyalarda saklanan veri arasında sıkı bir ilişki vardır. Herhangi bir dosyadaki fiziksel format (biçim) değişikliği bu dosyaya erişen tüm programlarda değişiklik gerektirecektir. Örneğin, ABD posta hizmetlerinde posta kodları 5 haneden 9 haneye çıkarıldığında binlerce bilgisayar programının yeniden organize edilmesi gerekmiştir.

#### a) Bordro Dosyası

Sosy. Güv. No	Adı	Saat Ücreti (\$)	Özel Sigorta	Bölüm
2314232	Ayşe	2,30	Yok	123
1242343	Fatma	4,20	Var	342
2341332	Selden	6,45	Var	323
*	*	*	*	*

#### b) Personel Dosyası

Sosy. Güv. No	Adı	Bölüm	Doğum Tarihi	İşe Giriş Tarihi
2314232	Ayşe	123	*	*
1242343	Fatma	342	*	*
2341332	Selden	323	*	*
*	*	*	*	*

Tablo 7.2 – Veri Tekrarlılığı ve Dosyalar Arası Bağlantı Kopukluğu

### 7.1.3.2 Veritabanı (VT) yaklaşımı

Bu yaklaşımda, her bir uygulama, birleştirilmiş veri dosyalarını kullanmaktadır. Yani veri dosyaları birleştirilmiştir. Bu yaklaşım, verinin birden fazla programca kullanılmasına izin veren bir yaklaşımdır. Şekil 7.3. bu yaklaşımı göstermektedir.

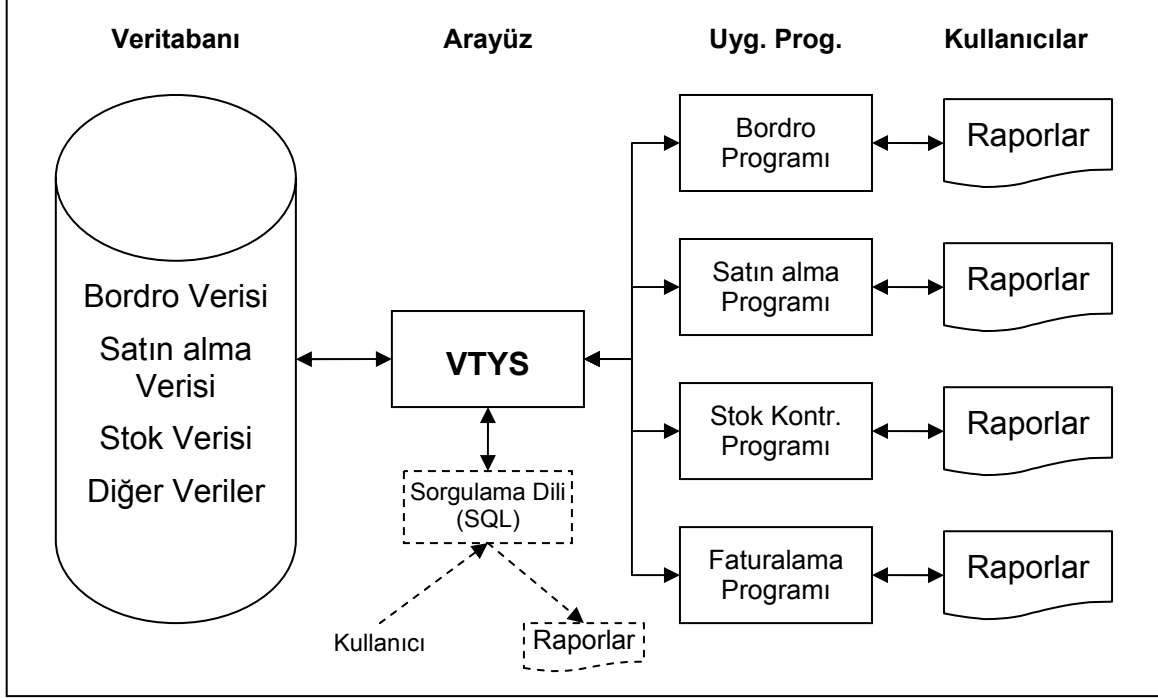
Veri yönetiminde VT yaklaşımın kullanmak için ilave bir yazılım yani VTYS gerekir. VTYS, bir organizasyonun, veriyi merkezileştirmesine, onları etkin bir şekilde idare etmesine ve saklanmış veriye uygulama programlarınca erişilmesine olanak tanıyan bir yazılımdır. VTYS, uygulama programları ile fiziksel veri tabanı arasında bir arayüz olarak görev yapmaktadır.

Bir kullanıcı veritabanındaki verilere ulaşmak ya da veritabanında güncelleme yapmak için dolaylı yöntem olarak uygulama programlarını kullanabilecekleri gibi doğrudan bir sorgulama dilini de kullanabilirler (SQL gibi).

VT yaklaşımı sayesinde geleneksel yöntemde karşılaşılan dezavantajlar ortadan kalkmıştır. *Veri tekrarlılığı* bütünleşik yapı sayesinde en aza inmiştir. *Dosyalar arası*

*kopukluk* ilişkisel yapı sayesinde giderilmiştir. *Program/veri bağımlılığı* da veritabanını organize eden ve yöneten ayrı bir yapı olan VTYS sayesinde ortadan kaldırılmıştır.

Bu yaklaşımın dezavantajı olarak kabul edilebilecek birkaç husus şunlardır. Böyle bir sistemin kurulmasının satın alma ve işletme masrafları çok yüksektir. Ayrıca veritabanının kullanımı, koordinasyonu ve yürütülmesi özel yetişmiş personel gerektirir. Bir diğer husus da veri tabanında meydana gelecek bir aksaklığın birden fazla uygulama programının etkileme olasılığıdır.



Şekil 7.3 – Veritabanı Yaklaşımı

## 7.2. Veritabanı Modelleri

Veritabanlarında, veri elemanları arasındaki ilişkinin mantıksal olarak yapılanması açısından uygulanan 3 farklı model vardır:

- i) Hiyerarşik (Ağaç) Yapılar
- ii) Ağ Yapılar
- iii) İlişkisel Yapılar

Bu derste, günümüzde en çok kullanılan ve avantajları diğerlerine göre daha fazla olan İlişkisel yapılar üzerinde durulacaktır. Hiyerarşik ve ağ yapılar hakkında da kısa bir bilgi vermek uygun olacaktır.

### 7.2.1 Hiyerarşik (Ağaç) Yapılar

Verileri, organizasyon şemasında olduğu gibi yukarıdan aşağıya doğru organize eden veritabanı yapısıdır. İlişkilerin bire-çok yapıda olması ağaç yapıları ayıran özelliğidir.

### 7.2.2 Ağ Yapılar

Ağ yapılar, ağaç yapısının bire-çoklu ilişkiden çoka-çok biçiminde genişletilmiş şeklidir. Belli bir veri elemanına erişilebilmesi birden fazla yol vardır.

### 7.2.3 İlişkisel Yapılar

İlişkisel veritabanı yapısı, veritabanına kayıt edilen bilgilerin belli kurallara uymasını içermektedir. Bu yapıya göre yapılanmış veritabanında tüm veri elemanları, satır ve sütunlardan oluşan basit tablolarda yer almaktadır. İlişkisel yapılar, verilerin tablolar, tablolar içinde sütunlar (alanlar) ve bir birincil anahtar kullanılarak depolandığı yapılardır. Tablolarda ortak sütun adları kullanılarak tablolar arası ilişkiler tanımlanır.

Bu yapının avantajı tam bir esneklik sağlaması ve gereksiz verinin barındırılmamasıdır. Bu yapı en popüler veritabanı yönetim yaklaşımıdır. Örnek olarak Oracle, MS SQL Server, IBM DB2, Sybase verilebilir.

İlişkisel yapıyı kullanan veritabanlarının temel bazı özellikleri vardır. **Sorgulama** özelliği bir uygulama programı kullanılmaksızın bir üst düzey dilin özel komutları ile veri tabanından bilgi çekilmesini sağlar. **Raporlama** özelliği sorgulama sonuçlarının sorgulama sonuçlarının ekran görünüşlerini ya da rapor formatının düzenlenmesini sağlar. İlişkisel veri tabanı tasarımında en fazla kullanılan tasarım aracı ilerleyen bölümlerde anlatılacak olan Birim İlişki Diyagramlarıdır (Entity Relationship Diagrams – ERD).

İlişkisel veritabanlarındaki veri yerleşimi ile geleneksel kayıt yerleşimi arasındaki fark tablo 7.3 ve tablo 7.4'te incelenebilir. Bu tablolarda kayıt tekrarını önlemek amacıyla büyük bir öğrenci ders kayıt dosyası etkin şekilde tablolara bölünmüştür.

Öğrenci_No	Adı_Soyadı	Not_Ort	Ders_Kodu	Ders_Adı	Not	Öğr. Gör.
2011029	İlker Dayı	2.67	ENM302	Sistem Analizi	A	F. Yegül
2011029	İlker Dayı	2.67	ENM314	İş Etüdü	B	Y. Babayev
2011312	Dilek Gülen	2.89	ENM302	Sistem Analizi	B	F. Yegül
2011134	Tuğrul Keleş	3.12	ENM412	Pazarlama	A	Y. Karaman
2011134	Tuğrul Keleş	3.12	ENM307	Kalite Kontrol	C	S. Kurt

Tablo 7.3 – Geleneksel Kayıt Yerleşimli bir Öğrenci Kayıt Dosyası

Öğrenci_No	Adı_Soyadı	Not_Ort
2011029	İlker Dayı	2.67
2011312	Dilek Gülen	2.89
2011134	Tuğrul Keleş	3.12

a) Öğrenci Dosyası

Ders_Kodu	Ders_Adı	Öğr. Gör.
ENM314	İş Etüdü	Y. Babayev
ENM302	Sistem Analizi	F. Yegül
ENM412	Pazarlama	Y. Karaman
ENM307	Kalite Kontrol	S. Kurt

b) Ders Dosyası

Öğrenci_No	Ders_Kodu	Not
2011029	ENM302	A
2011029	ENM314	B
2011312	ENM302	B
2011134	ENM412	A
2011134	ENM307	C

c) Kayıt Dosyası

Tablo 7.4 – İlişkisel Veri Tabanı

Tablo 7.4'te Öğrenci dosyası, Kayıt dosyasıyla Öğrenci\_No alanı vasıtasıyla, Kayıt dosyası da Ders dosyasıyla Ders\_kodu alanı vasıtasıyla ilişkilendirilmiştir. Böyle tasarlanmış bir veri tabanından sorgulama yoluyla rapor çekmek mümkündür. Örneğin 'Sistem Analizi' dersini alan öğrencilerin Öğrenci\_No, Adı\_Soyadı, Ders\_Kodu ve Notlarını bir tabloda görmek mümkündür.

Öğrenci_No	Adı_Soyadı	Ders_Kodu	Not
2011029	İlker Dayı	ENM302	A
2011312	Dilek Gülen	ENM302	B

Tablo 7.5 – Bir Sorgulama Sonuç Raporu

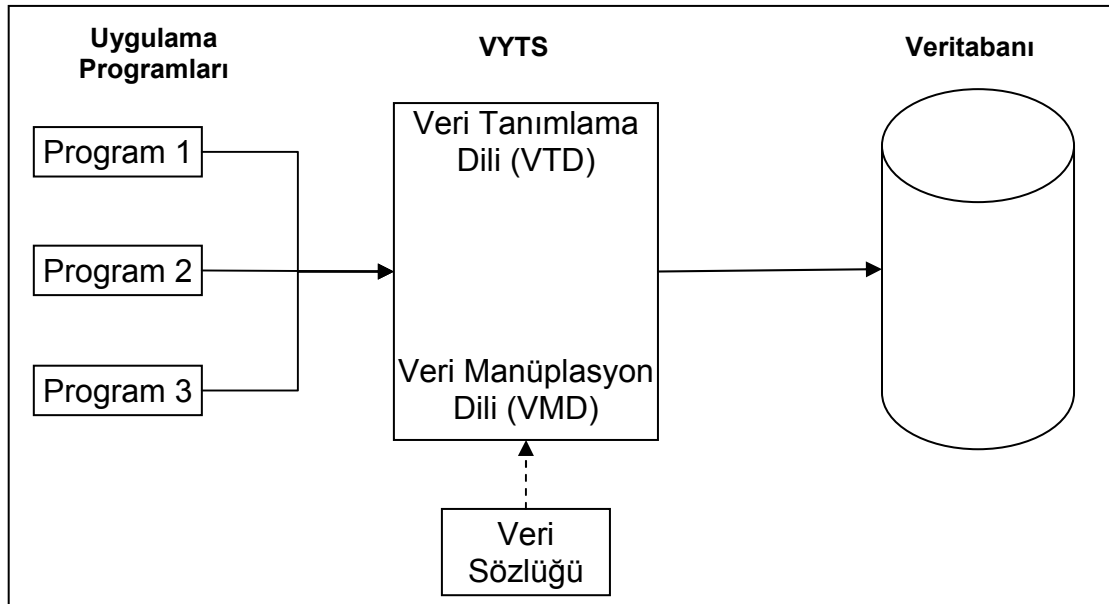
### 7.3. VTYS Bileşenleri

Bir VTYS, 3 temel bileşene sahiptir:

- 1) Veri Tanımlama Dili (VTD) (Data Definition Language)
- 2) Veri Manüplasyon Dili (VMD) (Data Manipulation Language)
- 3) Veri Sözlüğü (VS) (Data Dictionary)

Veri Tanımlama Dili, veritabanının yapısı ve içeriğini belirlemek için programcılar (VT Yöneticisi) tarafından kullanılan formal bir dildir. VTD, her kaydın fiziksel karakteristiklerini (kayıttaki alanlar, alanın tipi, uzunluğu gibi) tanımlamak için kullanılır.

Veri Manüplasyon Dili, veritabanındaki verinin idaresi için programcılar ve son kullanıcılar tarafından kullanılan VTYS ile ilgili bir dildir. Bu dil, veri tabanındaki kayıtların silinmesi, değiştirilmesi, ekranda gösterilmesi ve sıralanması gibi işleri yerine getirir. En bilinen VMD, SQL (Structured Query Language) dir.



Şekil 7.4 – VTYS'nin Bileşenleri

### 7.3.1 Veri Sözlüğü (VS) Kullanma

Veri sözlüğü, bir veritabanında muhafaza edilen veriyle ilgili bilginin tanımları ve karakteristiklerinin saklanması ve organize edilmesini sağlayan bir dosyadır. VS, veritabanında yer alan tüm verilerin bir envanterini oluşturarak önemli bir veri yönetim aracı olarak hizmet eder. Örneğin, bir kullanıcı, tüm firma tarafından tutulan bilgilerin neler olduğunu görmek ya da sadece pazarlama için hangi verilerin tutulduğunu belirlemek için VS'yi kullanabilir.

VS'nin derlenmesi bir veritabanı tasarlama işlemindeki ilk adımlardan biridir. VS'nin kullanılması;

- Terim ve veri elemanları için standart bir tanımlama sağlar.
- Programcılara, program tasarlama ve yazmada yardımcı olur.
- Sistem özelliklerini doküman eder.
- Veritabanı düzeltmelerini kolaylaştırır.
- Veri tekrarını azaltır; veri elemanlarında sağlanan standartlaşma veri tekrarlama hatalarını ortadan kaldıracaktır.

Veri sözlüğü için 5 tip giriş yapılabilir; (1) veri akışları, (2) veri depoları, (3) proses, (4) veri yapıları, (5) veri elemanları.

#### 7.3.1.1 Veri akışı sözlük girişi

Bir Veri Akış Diyagramındaki her bir veri akışını tanımlar. Bu akışın nereden nereye olduğu, verinin yapısının ne olduğu (kompozit [K] yada elemanter [E]) gibi bilgileri açıklar. Bazı örnekler şekil 7.5'te verilmiştir.

Veri Akış Sözlük Girişi	
<b>Veri Akış Adı</b>	: SATIŞ SİPARİŞLERİ
<b>Tanım</b>	: Müşterinin satın almak istediği ürünler ve her biri için gereken miktarı gösteren ve müşteri tarafından doldurulan dokümanlar
<b>Nereden</b>	: 1.0 Mektubu Aç Prosesi
<b>Nereye</b>	: 2.0 Satış Sipariş Verisini Gir
<b>Veri Yapıları</b>	: Satış Sipariş Kaydı (K)
<b>Açıklama</b>	: .....

Veri Akış Sözlük Girişi	
<b>Veri Akış Adı</b>	: FATURA
<b>Tanım</b>	: Müşteriye fatura edilecek doküman için gerekli bilgiler
<b>Nereden</b>	: 3.1 Faturayı hazırla
<b>Nereye</b>	: 3.2 Fatura numarası ata
<b>Veri Yapıları</b>	: Fatura Detayları (K) Müşteri Detayları (K)
<b>Açıklama</b>	: .....

Şekil 7.5 – Veri Akış Sözlük Girişi Örnekleri

### 7.3.1.2 Veri Deposu Sözlük Girişi

Bu giriş, VAD'daki her bir benzersiz veri deposunu tanımlar. Eğer VAD'da aynı veri deposu birden fazla varsa tek veri deposu girişi kullanılır. Veri depo adı VAD'da görülen ile aynıdır. Bir örnek şekil 7.6'da verilmiştir.

Veri Deposu Sözlük Girişi	
<b>Veri Depo Adı</b>	: SATIŞ SİPARİŞ FORM DOSYASI
<b>Tanım</b>	: Satış sipariş formlarının saklandığı arşiv dosyasıdır.
<b>Veri Yapıları</b>	: Satış Sipariş Kaydı
<b>Miktar</b>	: Yaklaşık günde 140 kayıt
<b>Erişim</b>	: Sipariş Bölümü Personeli
<b>Açıklama</b>	: .....

Şekil 7.6 – Veri Deposu Sözlük Girişi

### 7.3.1.3 Proses Sözlük Girişi

Bu giriş VAD'daki proseslerin girdilerini, ilgili proseste geliştirilen mantığı ve proses çıktılarını tanımlar. Şekil 7.7'de bir örnek verilmiştir.

Proses Sözlük Girişi	
<b>Proses</b>	: 2.0 Satış Sipariş Verisini Gir
<b>Girdi</b>	: Satış Siparişleri
<b>Proses tanımı</b>	: ..... yap, eğer değilse ..... yap vb.
<b>Çıktı</b>	: Girilmiş Satış Siparişleri

Şekil 7.7 – Proses Sözlük Girişi

### 7.3.1.4 Veri Yapısı Sözlük Girişi

Bu giriş, veri deposu ve veri akış formlarında listelenmiş her bir yapı için gerçekleştirilir. Veri akışları ve veri depoları, veri yapılarıdır. Eğer veri hareket ediyorsa onlara veri akışları, hareket etmiyorsa veri depoları denmektedir. Veri yapısı sözlük girişinde bulunan veri elemanlarının yanında (K) içerenlerinin her biri için bir veri yapısı sözlük girişinin yapılması gereklidir. Şekil 7.8'de bir örnek verilmiştir.

Veri Yapısı Sözlük Girişi	
<b>Yapı Adı</b>	: Satış Sipariş Kaydı
<b>Tanım</b>	: Müşterinin mal siparişi için kullandığı satış sipariş formu
<b>Veri Elemanları</b>	: MUSTERI_NO (E) MUSTERI_SIP_NO (E) MUSTERI_SIP_TAR (K) *KALEM_NO (E) *KALEM_MIKT (E) *KALEM_BIRIM_FYT (E)
<b>Açıklama</b>	: * olanlar, her bir kalem kaydı için oluşur

Veri Yapısı Sözlük Girişi	
<b>Yapı Adı</b>	: Fatura Detayları
<b>Tanım</b>	: Müşteriye aldığı mal karşılığı verilen belge
<b>Veri Elemanları</b>	: SIPARIS_NO (E) FATURA_NO (E) FATURA_TARIHI (K) TASIMA_GIDERLERI (E) KALEM_DETAYLARI (K) MIKTAR (E) VERGI (E)
<b>Açıklama</b>	:

Şekil 7.8 – Veri Yapısı Sözlük Girişi

### 7.3.1.5 Veri Elemanı Sözlük Girişi

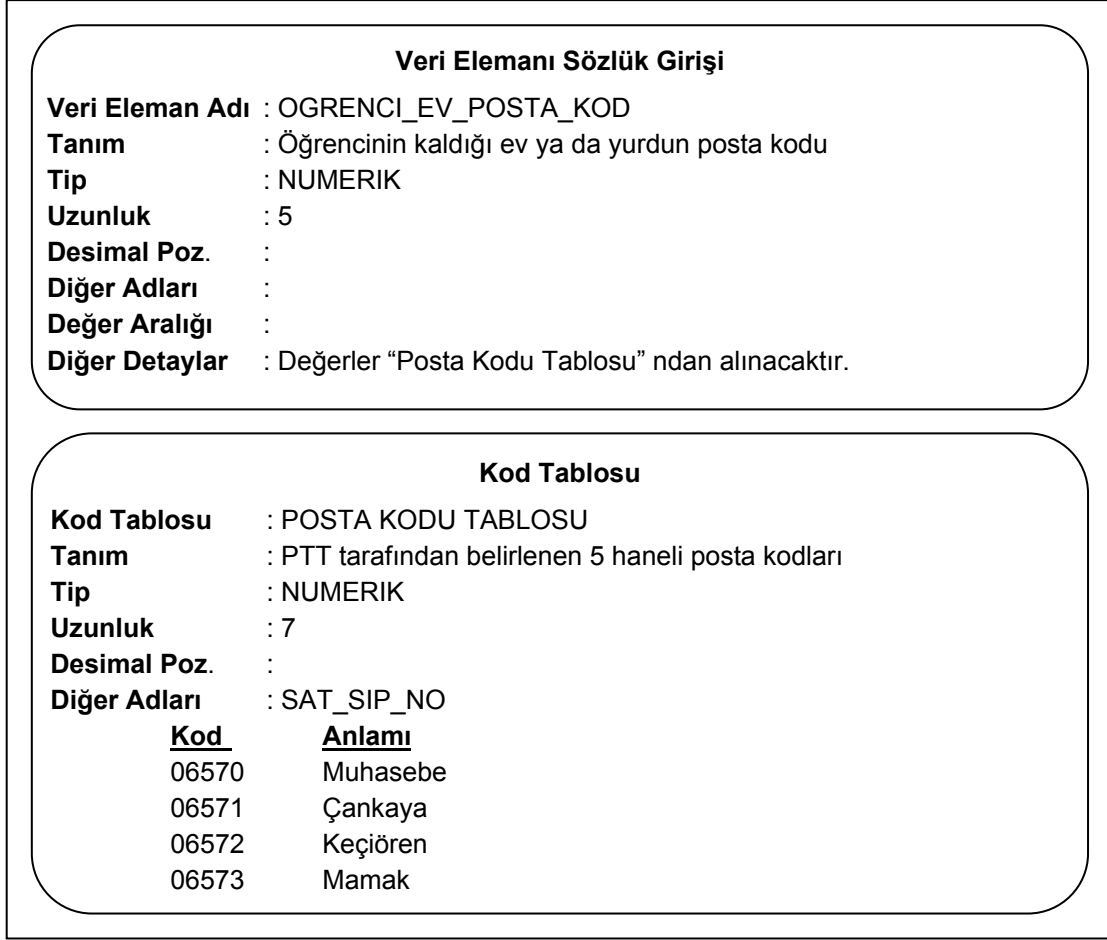
Bu giriş, veri yapılarının tamamındaki her bir veri elemanının tarifi için kullanılmaktadır. Eğer veri elemanı, sadece önceden belirlenmiş değerler listesinden bir değer içeriyorsa, o zaman bu değerleri ihtiva eden bir kod tablosu da oluşturulabilir. Veri elemanı sözlük girişleriyle ilgili örnekler şekil 7.9 ve 7.10'da verilmiştir.

Veri Elemanı Sözlük Girişi	
<b>Veri Eleman Adı</b>	: MUSTERI_NO
<b>Tanım</b>	: Müşteriyi tanımlayan numara
<b>Tip</b>	: NUMERİK
<b>Uzunluk</b>	: 6
<b>Desimal Poz.</b>	:
<b>Diğer Adları</b>	: MSTR_NO
<b>Değer Aralığı</b>	: 0001-9999
<b>Diğer Detaylar</b>	:

Veri Elemanı Sözlük Girişi	
<b>Veri Eleman Adı</b>	: SATINALMA_SIPARIS_NO
<b>Tanım</b>	: Dış tedarikçiye verilen her bir siparişi tanımlayan numara
<b>Tip</b>	: ALFANUMERİK
<b>Uzunluk</b>	: 7
<b>Desimal Poz.</b>	:
<b>Diğer Adları</b>	: SAT_SIP_NO
<b>Değer Aralığı</b>	:
<b>Diğer Detaylar</b>	:
<b>Belirli Değerler Listesi (Varsa)</b>	
<b>Önek</b>	<b>Anlamı</b>
MU	Muhasebe
SA	Satış
PE	Personel
AG	Araştırma geliştirme
<b>Diğer Detaylar</b>	: Satınalma sipariş numarası 5 basamaklı sayıdır ve önek içerir.

Şekil 7.9 – Veri Elemanı Sözlük Girişi



Şekil 7.10 – Veri Elemanı Sözlük Girişi ve Kod Tablosu

#### 7.4. İlişkisel Veritabanı Tasarımı

VAD’lar bir bilgi sisteminin modelini oluştururlar. VS ise, VAD’ı destekler. VAD’da bulunan veri depoları, bilgi sisteminde bilgisayar dosyalarına yerleştirilecek olan veri birimlerini ifade etmektedir. Bu dosyaların oluşturulmasındaki ilk adım, verinin mantıksal görünümünü gösteren veri modelleri oluşturmaktır. İlişkisel veritabanı tasarımında en fazla kullanılan yapısal tasarım (modelleme) aracı Birim İlişki Diyagramlarıdır (BİD, Entity Relationship Diagrams – ERD).

##### 7.4.1 Birim İlişki Diyagramları

Birim İlişki Diyagramları; veritabanı tasarımında kullanılan ve birimler arası ilişkileri gösteren diyagramlardır.

BİD oluşturulurken birimlerin (entity) ve onlarla ilgili özelliklerin (attributes) belirlenmesi gerekir. BİD’de VAD’da yer alan veri depolarının içinde yer alan veriler gösterilir. Bu diyagramlar, bir veritabanı için referans dokümanlar olarak kullanılabilir. BİD, kullanıcılar ve bilişim uzmanların ortak çalışması sonucunda oluşturulur.

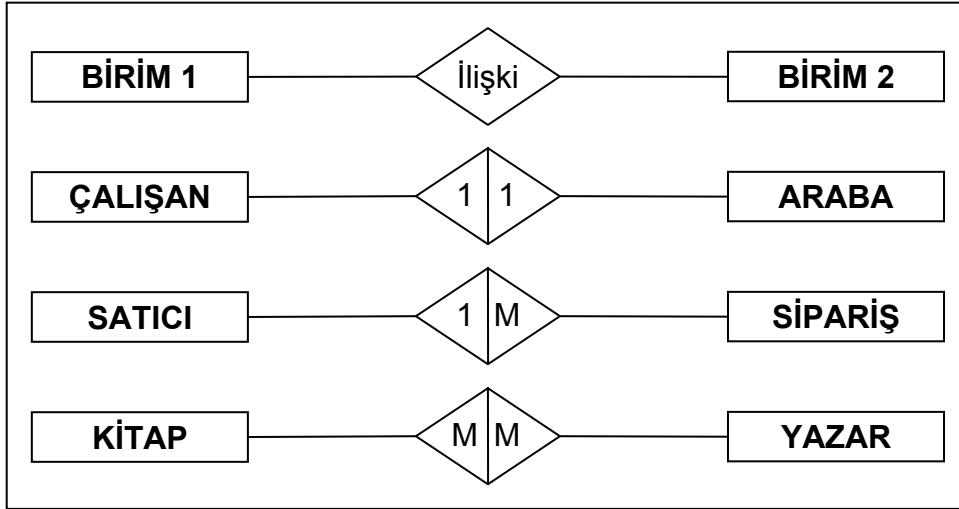
Bir **birim**, bir çevresel veri ya da bir kaynak olabilir. Örneğin, kişi, yer, müşteri, depo, parça, ürün, ekipman, sipariş vb hepsi birer birimdir. Birimler, BİD’de diktdörtgen sembolüyle ifade edilirler. Özellik ise herhangi bir birime ait durumlardır. Örneğin, müşteri no, müşteri adı, müşteri adresi vb. durumlar, müşteri biriminin özellikleri olarak ifade edilebilir.

Bir **ilişki**, iki ya da daha fazla veri birimi arasındaki mantıksal etkileşimdir. BİD, hem birimleri hem de ilişkileri gösterir. İlişkiler karo (eşkenar dörtgen) sembolü ile gösterilir.

İki birim arasında genelde 3 tip ilişkiden söz edilebilir:

- i) Bire-bir (one-to-one) ilişki (1:1)
- ii) Bire-çok (one-to-many) ilişki (1:M)
- iii) Çoka-çok (many-to-many) ilişki (M:M)

Şekil 7.11'de bu ilişkiler sembollerle gösterilmiştir.



Şekil 7.11 – Birimler ve İlişkiler

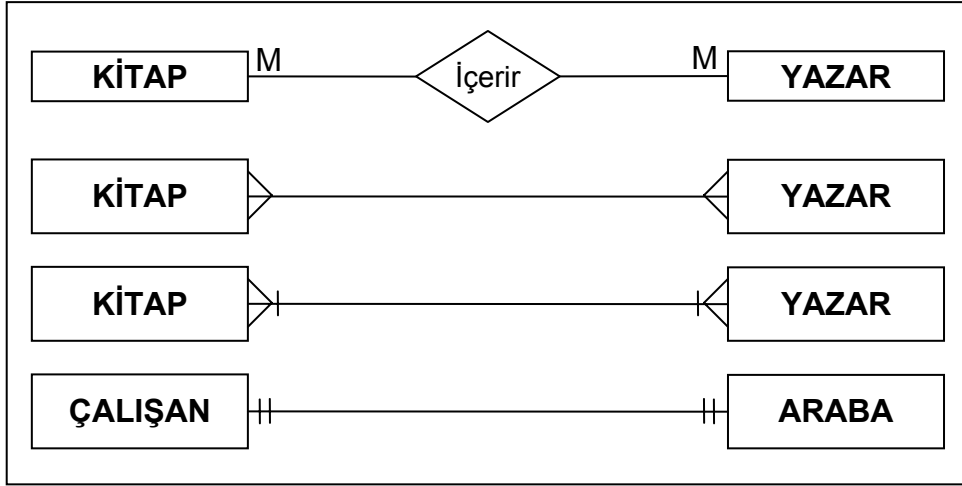
Şekil 7.11'de verilen ilişki diyagramlarını kısaca açıklayalım.

Bire-Bir tanımlanan Çalışan ve Araba ilişkisinde, her bir çalışana sadece bir araba tahsis edilmekte ve bunun tersi olarak da her bir araba sadece bir çalışana tahsis edilmektedir. Diğer örnekler kişi ile kimlik, zaman kartı, bordro fişi arasında olabilir.

Satıcı ile sipariş arasındaki Bire-Çok ilişki, bir satıcının (satış elemanı) birden fazla sipariş ile ilgili olabildiği, buna karşılık bir siparişin sadece bir satıcı tarafından alınabildiğini ifade eder. Başka örnek olarak, ürün ile parça, bölüm ile personel verilebilir.

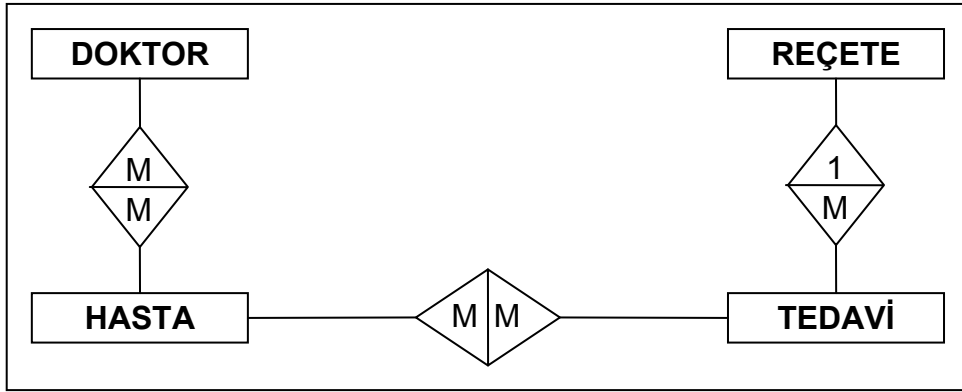
Son olarak Kitap ve Yazar arasındaki ilişki Çoka-Çok ilişkiye örnektir. Burada bir kitap birden fazla yazar tarafından yazılmış olabileceği gibi bir yazar da birden çok kitap yazmış olabilir. Diğer örnekler proje ile çalışan, müşteri ile ürün arasındaki ilişkiler olabilir.

BİD için kullanılan farklı gösterim şekilleri de vardır. Bunlar şekil 7.12'de gösterilmiştir.



Şekil 7.12 – BİD'in Farklı Gösterimleri

Şekil 7.13'te Doktor, Hasta Tedavi ve Reçete birimleri için örnek bir BİD verilmiştir.



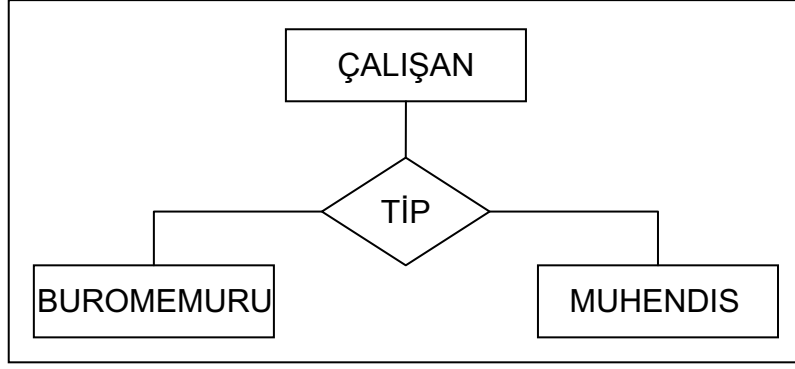
Şekil 7.13 – Örnek BİD

Bu örnekte verilen ilişkiler bir başka sistem için başka olabilir. Örneğin başka özel bir sistemde Doktor ile Hasta arasındaki ilişki bire-bir olabilir.

Bir veri birimi için özelliklerin kümesi farklı olabiliyorsa, o zaman nesne tipi, Alttıplerden oluşan bir Üsttip ile gösterilebilir. Nesne tipi CALISAN'ın dört adet özelliğe sahip olduğu (CALISAN\_NO, CALISAN\_ADI, CALISAN\_ADRES, CALISAN\_MAAS) bir sistem düşünelim. Firma, büro memurlarıyla ilgili yazma yetenekleri, yazma hızı vb. bilgileri bilmek, mühendisler ile alakalı olarak da her birinin sahip olduğu profesyonellik belgelerini, onların profesyonel birliklere üyeliklerini izlemek istiyor olsun.

Bu durumda BUROMEMURU ve MUHENDIS gibi iki veri birimi oluşturulmasından ziyade, BİD için BUROMEMURU ve MUHENDIS alttıplerinden oluşan bir CALISAN üsttipi oluşturmak uygun olur.

Burada üsttipteki birimlerin özellikleri aynı karakteristikleri paylaşacaklardır. Örneğin her bir çalışan CALISAN\_NO, CALISAN\_ADI, CALISAN\_ADRES, CALISAN\_MAAS gibi özelliklere sahip olacaktır. Alttıplerin özellikleri ise birbirinden farklı olacaktır. Bu örneğin şekilsel gösterimi şekil 7.14'te verilmiştir.



Şekil 7.14 – Üst ve Alt Tipler

#### 7.4.2 Normalizasyon için Yedi Adım Metodu

Normalizasyon için yedi adım metodu Ek3'te anlatılmıştır.

Normalizasyon için Yedi Adım Metodu:

*Ek3 – Normalizasyon için Yedi Adım Metodu / YBS Kitabı (Sayfa 144,158)*