

# 1

## Amaçlarımız

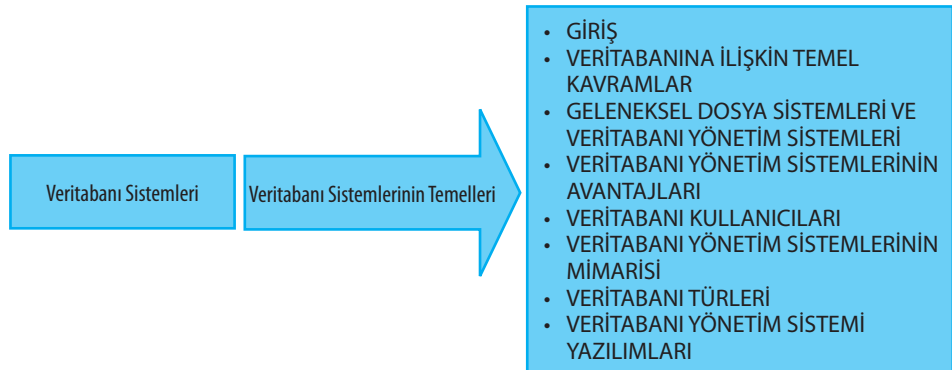
Bu üniteyi tamamladıktan sonra;

- Veritabanına ilişkin temel kavramları tanımlayabilecek,
- Geleneksel dosya sistemleri ile veritabanı yönetim sistemlerini karşılaştırabilecek,
- Veritabanı kullanıcılarını sınıflandırabilecek,
- Veritabanı yönetim sistemlerinin mimarisini açıklayabilecek,
- Veritabanı türlerini ve yaygın olarak kullanılan veritabanı yönetim sistemi yazılımlarını sıralayabilecek bilgi ve becerilere sahip olacaksınız.

## Anahtar Kavramlar

- Veri
- Veritabanı
- Veritabanı Yönetim Sistemi
- Veri Modeli
- Veritabanı Şeması
- Veri Tanımlama Dili
- Veri İşleme Dili
- Sorgulama Dili

## İçindekiler



# Veritabanı Sistemlerinin Temelleri

## GİRİŞ

Günümüzde küçük ya da büyük tüm işletmeler faaliyetlerini yerine getirebilmek, rakipleriyle yarışabilmek ve varlıklarını başarılı bir biçimde sürdürebilmek için bilgiye ihtiyaç duyarlar. Bu nedenle bilgi, günümüzde üretim faktörleri içinde yer almakta ve sermaye, hammadde, enerji ve iş gücü olarak sıralanan geleneksel üretim faktörleri kadar önem taşımaktadır. Bununla birlikte bilginin istenilen amaca hizmet edebilmesi için zamanında elde edilebilmesi ve amaca uygun olması gerekir. İşletmelerin faaliyetlerini düzenli ve etkin bir biçimde yerine getirebilmelerinde hayati bir öneme sahip olan bilginin zamanında elde edilebilmesi ve ilgili kişi ve birimlere ihtiyaç duyulduğu anda iletilebilmesi ise bilgi sistemleri aracılığıyla gerçekleştirilir. Bilgi sistemleri; ilgili verilerin toplanması, bu veriler üzerinde birtakım işlemler uygulanarak verilerin yararlı bilgilere dönüştürülmesi ve ilgili kişi ve birimlere iletilmesini, belirli bir düzen içinde gerçekleştiren sistemlerdir. İşletmelerde verilerin toplanması ve işlenmesi için kullanılan araçlar, teknikler, yaklaşımlar geçmişle günümüz arasında farklılık gösterse de bilgi sistemlerinin taşıdığı önem aynıdır. Teknoloji ve bilgisayar sektöründeki gelişmelere paralel olarak günümüzde bilgi sistemleri, bilgisayar donanımı, yazılım, veri kaynakları, ağ teknolojileri ve insan bileşenlerinden oluşmaktadır. Bu bileşenlerden veri kaynakları, işletmede ihtiyaç duyulan bilgiyi üretmek amacıyla kullanılacak verinin saklandığı (depolandığı) ve ihtiyaç duyulduğu anda erişilebildiği bileşeni ifade etmektedir. Bu kitapta ele alınan veritabanı sistemleri de kısaca veri kaynaklarının yönetiminden sorumlu sistemlerdir. Veritabanı sistemlerine ilişkin ayrıntılı bilgilerin yer aldığı izleyen ünitelerden önce bu ilk ünite veritabanı sistemlerine ilişkin temel kavramlar ve genel bilgiler ele alınmıştır.

## Veritabanının Tarihçesi

Bir konuyla ilgili verilere hızlı ve kolay biçimde erişebilme isteği, verilerin düzenli biçimde saklanması ihtiyacını doğurmuş, dolayısıyla veritabanı olgusunun bu adla olmasa bile geçmişten günümüze kadar bir biçimde hayatımızda yer almasına neden olmuştur. Geçmişle günümüz arasındaki fark kullanılan araç, gereç, teknik ve yaklaşımlarda ortaya çıkmaktadır. Geçmişte veriler fiziksel olarak, kağıt, defter, dosya kullanımı ile saklanmaktaydı. Bu anlamda yalnızca ev telefonlarının olduğu geçmişte, iletişim kurmak istediğimiz kişilerin ad, soyad ve telefon numaralarının yer aldığı ve klasik defter biçiminde düzenlediğimiz telefon rehberleri örnek olarak verilebilir. Benzer biçimde devlet kurumları, hastaneler ve ticari kuruluşlar da ihtiyaç duydukları verileri kağıt dosyalar biçiminde ve belirli bir düzende arşivlemekte ve gerekli olduğunda bu arşivlerde arama yaparak istenen

veriye erişmekteydi. Bu tür sistemlerde, arşivde istenen bilgiye erişmek için belirli kurallar ve düzenlemeler uygulanmaktaydı. Kütüphanelerde herhangi bir kitaba erişmek için kullanılan kitap künye bilgisi bu tür düzenlemelere örnek olarak verilebilir. Bu sistemlerin bazı temel prensipleri günümüzde hâlâ kullanılmaktadır. Düzenlenmesi ve saklanması gereken veri miktarının giderek artması ve bu verilerin fiziksel olarak uzun süre saklanabilmesi, korunabilmesi ve istendiğinde hızlıca erişilebilmesi kolay bir iş değildi. Söz konusu bu zorlukların üstesinden gelebilme istek ve ihtiyacı bilgisayar teknolojilerindeki gelişmelerin temel nedenlerinden biri olmuştur. Bilgisayar teknolojisinde bugün gelinen noktada çok büyük miktarda ve karmaşık yapıda olan verilerin saklanması ve yönetilmesi oldukça kolaylaşmıştır. Bu nedenle günümüzde veritabanı denilince akıllara bilgisayar ortamında oluşturulan veritabanları gelmektedir.

Bilgisayar kullanımının özel işletmeler için uygun maliyetli bir seçenek olmaya başladığı dönem olan 1960'larda veritabanlarının bilgisayarlar yardımıyla oluşturulması dönemi de başlamıştır. İzleyen kesimde 1960'lı yıllardan itibaren veritabanı uygulamalarının tarihsel gelişimine ilişkin kısa bilgilere yer verilmiştir.

**1960'lar:** 1960'lı yılların başında Charles Bachman tarafından IDS (Integrated Data Store-Bütünleştirilmiş Veri Depolama) adıyla ilk genel amaçlı veritabanı yönetim sistemi geliştirilmiştir. 1960'ların sonunda ise IBM tarafından IMS (Information Management System-Bilgi Yönetim Sistemi) adıyla ilk ticari VTYS (Veritabanı Yönetim Sistemi) geliştirilmiş ve bu yapı hiyerarşik veri modeline temel teşkil etmiştir (Vural ve Sağıroğlu, 2010, s.72).

1967'de CODASYL (CONference on Data SYstem Languages) grubu kullanıcı ve bilgisayar üreticileri için COBOL dili standartlarını belirlemiş ve bu standartlar ANSI (American National Standards Institute-Amerikan Ulusal Standartlar Enstitüsü) tarafından kabul edilmiştir. Bu standartlaştırma başarısından sonra CODASYL grubu veritabanı standartlarını oluşturmak için DBTG'yi (DataBase Task Group-Veritabanı Görev Grubu) kurarak grubu veritabanı standartlarını oluşturmak üzere görevlendirmiştir. DBTG yaptığı çalışmalar sonucunda ağ veri modelleri için dil özelliklerini, veritabanı şemalarını, veri tanımlama ve veri işleme dili gibi önemli veritabanı bileşenlerini tanımlamıştır.

1969'da Edgar F. Codd tarafından geliştirilen ilişkisel veritabanı modeli, insanların veritabanları hakkındaki düşüncelerini değiştirecek nitelikte bir gelişme olarak ortaya çıkmıştır. Modelde, veritabanının şeması veya mantıksal organizasyonu, fiziksel bilgi deposundan bağımsız hâle getirilmiş ve bu da veritabanları için standart bir prensip hâline gelmiştir.

**1970'ler:** 1974 ve 1977 yılları arasında iki önemli ilişkisel veritabanı sistemi prototipi oluşturulmuştur. Bunlardan biri UBC'de geliştirilen Ingres, diğeri ise IBM San Jose'de oluşturulan System R'dir. Ingres, QUEL diye bilinen sorgulama dili olarak kullanılmış ve Ingres Corp., MS SQL Server, Sybase, Wang's PACE, ve Britton-Lee gibi sistemlerin oluşumuna öncülük etmiştir. Diğer yandan System R, SEQUEL sorgulama dilini kullanmış ve SQL/DS, DB2, Allbase, Oracle, ve Non-Stop SQL'in gelişimine katkıda bulunmuştur. Ayrıca, İlişkisel Veritabanı Yönetim Sistemi (RDBMS) bu dönemde bilinen bir terim hâline gelmiştir.

Varlık-İlişki (Entity-Relationship) diye adlandırılan yeni bir veritabanı modeli, Peter Chen tarafından 1976 yılında önerilmiştir. Bu model, tasarımcıların mantıksal tablo yapısı yerine, veri uygulamasına odaklanmalarını mümkün kılmıştır.

**1980'ler:** Yapılandırılmış Sorgu Dili (SQL) standart sorgu dili hâline gelmiştir. Buna paralel olarak ilişkisel veritabanı sistemleri, bilgisayar satışlarındaki hızlı yükselişin veritabanı piyasasını canlandırmasıyla ticari bir başarıya dönüşmüştür. Bu da ağ ve hiyerarşik veritabanı modellerinin popüleritesinde büyük bir düşüşe yol açmıştır. Yine bu yıllarda DB2, IBM'in popüler veritabanı ürünü olmuştur. IBM PC'nin tanıtımı da birçok yeni veritabanı şirketinin kuruluşuna ve PARADOX, RBASE 5000, RIM, Dbase III ve IV, OS/2 Database Manager ve Watcom SQL gibi ürünlerin gelişimine yol açmıştır.

**1990'lar:** 1990'ların başında veritabanı endüstrisinde yaşanan kriz nedeniyle bu sektörde varlığını sürdüren şirketler krizi fırsat bilerek karmaşık veritabanı ürünlerini yüksek fiyatlarla satabilmişlerdir. Bu sıralarda, Oracle Developer, PowerBuilder, VB gibi uygulama geliştirme araçları piyasaya sunulmuştur. Ayrıca Access programı da 1990'lı yılların başında piyasaya sürülmüştür. Yine bu dönemde Nesne Veritabanı Yönetim Sistemleri (ODBMS) için prototipler de geliştirilmiştir. 1990'ların ortalarında internetin kullanılmaya başlanması veritabanı endüstrisinde hızlı bir büyümeye yol açmıştır.

1990'ların sonlarında online işletmelere artan yatırımlar nedeniyle Front Page, Active Server Pages, Java Servlets, Dream Weaver, ColdFusion, Enterprise Java Beans, ve Oracle Developer 2000 gibi internete bağlı veritabanı bağlayıcılarına olan talep miktarı belirgin olarak artmış, yine bu dönemlerde MySQL, Apache, vb. gibi açık kaynak kodlu veritabanı yönetim sistemi yazılımları kullanıma sunulmuştur.

**2000'ler:** İnternet endüstrisi 2000'lerin başında bir düşüş yaşamasına rağmen veritabanı uygulamaları büyümeye devam etmiştir. Web tabanlı işletmelerin artması ve kullanımlarının yaygınlaşması ile veritabanı yönetim sistemlerinin önemi giderek artmaya devam etmekte ve yeni uygulamaların geliştirilmesine neden olmaktadır. Günümüzde, veritabanı yönetim sistemi yazılımlarında lider kabul edilen üç işletme Microsoft, IBM ve Oracle olarak sıralanmaktadır.

## VERİTABANINA İLİŞKİN TEMEL KAVRAMLAR

Günümüzde sıklıkla karşılaşılan bir kavram hâline gelen *veritabanı* çok genel anlamda, bir kurumun ihtiyaç duyduğu ve kullandığı veriler bütünüü ifade eder. Bu noktada öncelikle *veri* kavramı ve *veri* kavramı ile karıştırılan *bilgi* kavramı üzerinde durmak faydalı olacaktır.

*Veri* ham gözlemler, işlenmemiş gerçekler ya da izlenimlerdir. Bu gözlemler, gerçekler ya da izlenimler harf, rakam ya da çeşitli sembol ve işaretler yardımıyla temsil edilir. Birbirleriyle ilişkilendirilip yorumlanmadıkları sürece tek başlarına bir anlam ifade etmezler ve bu hâllerile karar verme konusunda da karar vericilere bir katkı sağlayamazlar.

*Bilgi* ise yalın tanımıyla verinin işlenmiş ve karar verme sürecine destek olacak duruma dönüştürülmüş biçimidir. Sözkonusu işleme ve dönüştürme süreci; veri üzerinde kaydetme, sınıflama, sıralama, hesaplama, özetleme, çoğaltma, analiz ve raporlama işlemlerinin uygulanması ile gerçekleştirilir. Bu işlemler sonucunda veri anlam kazanarak bilgiye dönüşmüş olur.

*Veritabanı*, (database) herhangi bir konuda birbiriyle ilişkili olan ve amaca uygun olarak düzenlenmiş, mantıksal ve fiziksel olarak tanımlanmış veriler bütünüdür. Bununla birlikte her düzenli veri topluluğunu veritabanı olarak tanımlamak da doğru değildir. Bu nedenle veritabanının özelliklerinin sıralanması faydalı olacaktır.

- Veritabanı herhangi bir kurumda birden fazla uygulamada ortak olarak kullanılabilen verilerden oluşur.
- Veritabanında sürekli niteliği olan veriler bulunur. Buna göre, girdi ya da çıktı verisi olan ya da kurum için sürekli bir anlam ifade etmeyen geçici veriler veritabanında yer almaz.
- Veritabanı, ortak kullanılan verilerin tekrarlanmasına izin vermeden çok amaçlı kullanılmasına olanak verir.
- Veritabanında saklanan veriler durağan nitelikte değişmez veriler değildir. Ekleme, silme ya da güncelleme işlemleri ile veritabanındaki veriler değiştirilebilir (Yarımağan, 2000, s.1).

*Veritabanı Yönetim Sistemi* (VTYS-Database Management System); veritabanı tanımlamak, veritabanı oluşturmak, veritabanında işlem yapmak, veritabanının farklı kullanıcı yetkilerini belirlemek, veritabanının bakımını ve yedeklemesini yapmak için geliştirilmiş programlar bütünüdür.

**Veritabanı Tanımlamak:** Veritabanında yer alacak verinin adı, tipi, uzunluğu gibi veri yapıları ve özelliklerinin belirlenmesidir.

**Veritabanı Oluşturmak:** Veritabanını yaratma ve veriyi depolama sürecidir. Verilerin depolanacağı fiziksel bellek alanının belirlenmesi ve verilerin bu alana aktarılmasını içerir.

**Veritabanında İşlem Yapmak:** Veritabanını izleme, veriler üzerinde sorgulama yapma, veritabanında gerekli olan değişiklikleri yaparak veritabanını güncelleme ve verilerden rapor elde etme işlevlerini içerir.

Bu özelliklerin dışında veritabanı, veriyi koruma, veriler arasında ilişki kurma, farklı kullanıcılara farklı yetki sınırları içinde veriye erişim imkânı sunma işlevlerini de yerine getirir.

Veritabanı ve veritabanı yönetim sisteminin birlikte oluşturduğu bütün ise *veritabanı sistemi* olarak ifade edilir.

Geçmişten beri veriye erişim amacıyla farklı yaklaşımlar kullanılmıştır. Bu yaklaşımlardan ikisi; *sıralı erişim* ve *doğrudan erişim* biçimindedir.

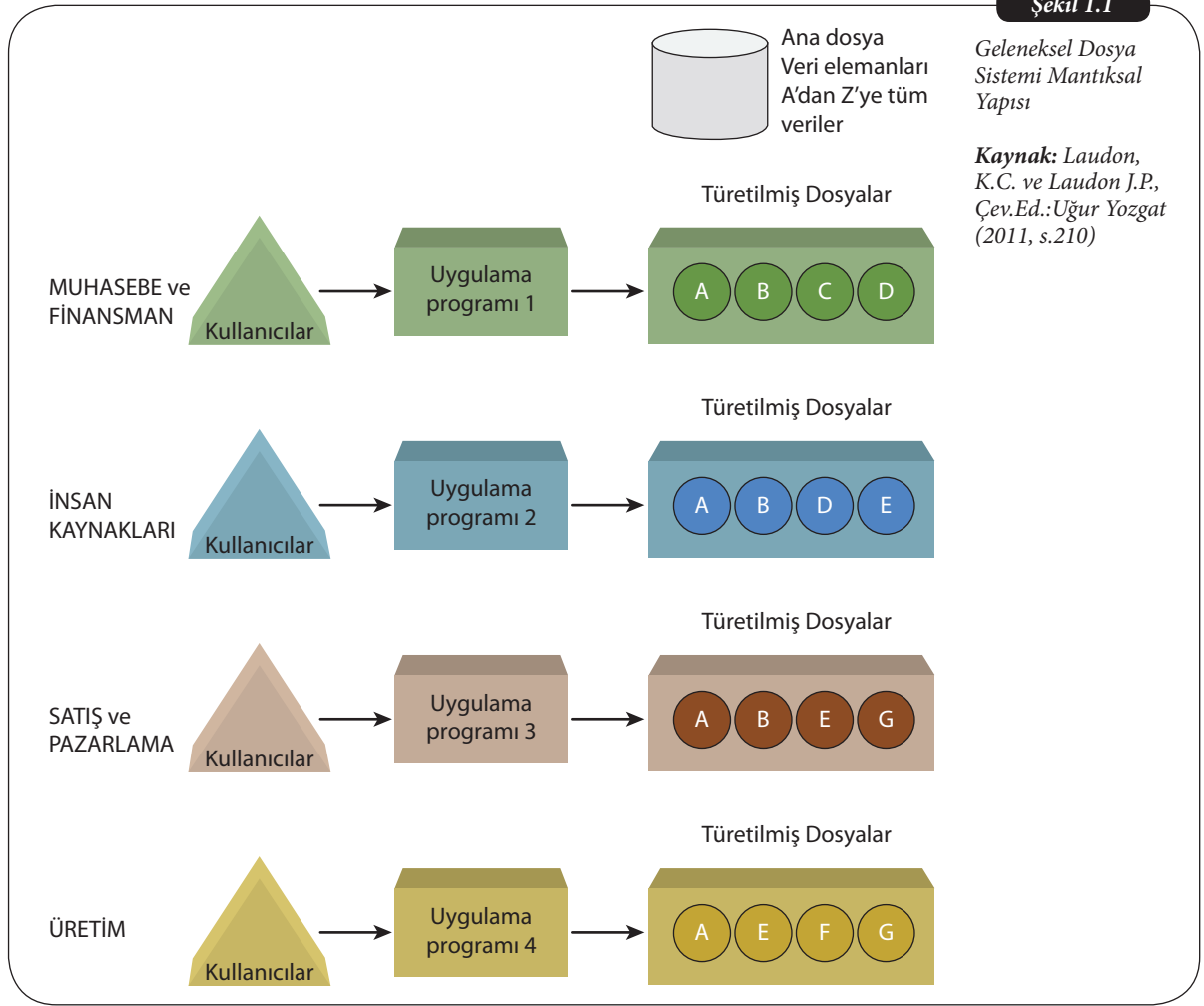
**Sıralı erişimde,** istenilen veriye ulaşıncaya kadar ilgili dosyadaki tüm verilerin sırayla okunması gerekir. Geçmişte kullanılan müzik kasetleri bu tür erişim kullanımına örnektir. Bu kasetlerde şarkı olarak dinlediğimiz müzik verileri sırayla çalınmaktaydı. Kasette yer alan 6 numaralı şarkıyı dinleyebilmek için ya ilk 5 şarkıyı dinlemek ya da kasetçaların ileri düğmesine basarak 5 şarkılık verinin fiziksel olarak ileri sarılması gerekirdi. Her iki durumda da 6. şarkıya gelebilmek için ilk 5 şarkının fiziksel olarak taranması işlemi yapıldı. Bu şekilde, dosyada yer alan tüm verilerin okunması zorunluluğu ve istenilen bilgiye anında ulaşılamaması sıralı erişimin dezavantajı olarak ortaya çıkmaktadır. Bir diğer örnek olarak öğrenci bilgilerinin yer aldığı bir dosyada, dosyanın en sonunda yer alan bir öğrenci bilgisine erişmek için dosyanın tamamının okunması gerekecektir. Buna rağmen sıralı erişimin gerekli olduğu bazı durumlar da bulunmaktadır.

**Doğrudan erişimde,** sıralı erişimin aksine istenilen veriye ulaşabilmek için o veriye kadar olan diğer tüm verilerin okunması gerekmemektedir. Bu erişim biçiminde adından da anlaşılacağı gibi istenilen veriye doğrudan erişim mümkündür. Bu erişim biçiminde, verilerin yer aldığı fiziksel adresler birer indeks numarası ile tanımlanıp bu indeks numaraları da ayrı bir dosya olarak saklanır. Herhangi bir veriye erişilmek istendiğinde, önce indeks dosyasından verinin yer aldığı adresi gösteren indeks numarası bulunur, daha sonra bu numaraya karşılık gelen fiziksel adrese doğrudan erişim sağlanır. Günümüzde kullanılan müzik CD'lerindeki erişim türü bu erişim türüne örnektir.

## GELENEKSEL DOSYA SİSTEMLERİ VE VERİTABANI YÖNETİM SİSTEMLERİ

Verilerin belirli bir düzen içinde saklanması ve ihtiyaç duyulduğunda erişilebilmesi amacıyla bilgisayar kullanımında geçmişten günümüze farklı yaklaşımlar benimsenmiştir. Veritabanı yaklaşımından önce kullanılan ve *geleneksel dosya sistemi* olarak ifade edilen yaklaşımda veriler bilgisayarda ayrı ayrı dosyalar biçiminde saklanmaktaydı. Birbiriyle ilişkili olan veriler bir dosyada, başka bir açıdan birbiriyle ilişkili veriler de başka bir dosyada yer almaktaydı. Bununla birlikte iki farklı dosya içinde aynı verinin yer alması da söz konusu idi.

Şekil 1.1



Şekil 1.1'den de izlenebileceği gibi, geleneksel dosya sisteminde ilgili kuruluşun her alt sistemi kendi faaliyetlerini gerçekleştirmek için o alt sisteme özgü uygulamalar ve veri dosyalarına ihtiyaç duyar. Geleneksel dosya sisteminde, tüm verilerin saklandığı bir ana dosyanın yanı sıra her alt sisteme özgü ve ana dosyanın altkütmesi olarak düşünülebilecek birçok alt dosya türetilir. Bu durum da verinin gereksiz yere tekrarına ve dolayısıyla bellek kapasitesinin boş yere kullanılmasına yol açar. Şekil 1.1 incelendiğinde A verisinin, işletmenin dört farklı alt sistemi olarak düşünülen tüm bölümlerinde ihtiyaç duyulan bir veri olduğu ve dolayısıyla bu bölümlerde türetilmiş tüm dosyalarda yer aldığı görülmektedir. Benzer biçimde B verisi üç farklı bölümün dosyalarında yer işgal etmektedir. F verisinin ise yalnızca üretim bölümünün ihtiyaç duyduğu bir veri olduğu ve diğer bölümlerin veri dosyalarında yer almadığı farkedilecektir. Bununla birlikte F verisi dışındaki diğer tüm verilerin tekrarlandığı görülmektedir. Aynı verinin birçok dosya içinde tekrarlanması veri yönetimiyle ilgili pek çok zorluğu da beraberinde getirecektir.

Zaman içinde artan veri miktarı, bu verinin depolanması için gerekli olan kapasite, veriye erişim ve işleme hızında yaşanan sıkıntılar geleneksel dosya sisteminin temel sınırlılıkları olarak ortaya çıkmıştır. Dolayısıyla bu sıkıntılar modern veritabanı sistemlerinin geliştirilmesini de gerekli hâle getirmiştir.

Bilgisayar tabanlı bilgi sistemleri kapsamında kullanılan ve farklı uygulamalar tarafından paylaşılan ortak verilerin düzenlenmesi, saklanması ve kullanılması amacıyla yönelik iki farklı yaklaşım vardır:

- Geleneksel dosya sistemi
- Veritabanı yönetim sistemi

Geleneksel dosya sistemlerinin oluşturduğu olumsuzluklar ve buna karşın veritabanı yönetim sistemlerinin sağladığı yararlar Tablo 1.1’de verilen biçimde özetlenebilir.

**Tablo 1.1**  
Geleneksel Dosya Sistemleri ile Veritabanı Yönetim Sistemlerinin Karşılaştırılması

Geleneksel dosya sistemlerinin sakıncaları	Veritabanı yönetim sistemlerinin üstünlükleri
<ul style="list-style-type: none"> <li>• Veri tekrarı ve veri tutarsızlığına yol açar.</li> <li>• Veri paylaşımına olanak vermez.</li> <li>• Uygulamalarda ihtiyaç duyulan değişikliklerin gerçekleştirilebilmesi için uzmanlık bilgisi gerektirir.</li> <li>• İstenilen veriye ulaşmada güçlükler bulunur.</li> <li>• Verilerin güvenliği ve gizliliği konusunda sorun yaşanır.</li> <li>• Veriler ve uygulamalarla ilgili belirli bir standart yoktur.</li> <li>• Verileri yedekleme ve kurtarma konusunda güçlükler yaşanır.</li> </ul>	<ul style="list-style-type: none"> <li>• Veri tekrarı ve veri tutarsızlığını önler.</li> <li>• Veri paylaşımına olanak verir.</li> <li>• Uzmanlık bilgisine ihtiyaç duyulmayacak derecede kullanım kolaylığı sağlar.</li> <li>• İhtiyaç duyulan veriye, tanımlanmış kullanıcı yetkileri kapsamında kolaylıkla erişilmesini sağlar.</li> <li>• Veri güvenliği ve gizliliğini güçlü bir biçimde yerine getirir.</li> <li>• Veriler ve uygulamalarla ilgili standart yapı ve kuralların olması kullanım kolaylığı sağlar.</li> <li>• Verileri yedekleme ve kurtarma konusunda kolaylık sağlayan programlar barındırır.</li> </ul>

VTYS’ler, sağladıkları avantajlara rağmen maliyet açısından geleneksel dosya sistemlerine göre dezavantajlıdır. VTYS için gerekli olan donanım, yazılım ve veritabanı eğitimi için başlangıç yatırımlarının yüksek olması, veri güvenliğini sağlama, aynı anda gerçekleştirilen işlemlerin kontrolü, veri kurtarma ve bütünlük fonksiyonları için gerekli olan sabit maliyetlerin yüksek olması VTYS’nin dezavantajlı yönlerini oluşturmaktadır.

Bu nedenle değişmesi hiç beklenmeyen basit ve iyi tanımlanmış veritabanı uygulamaları söz konusu olduğunda geleneksel dosya sistemlerini kullanmak daha avantajlı olacaktır.

## VERİTABANI YÖNETİM SİSTEMLERİNİN AVANTAJLARI

Tablo 1.1’de verilen özelliklere ilişkin ayrıntılı açıklamalar, veritabanı yönetim sistemlerinin üstünlüklerine vurgu yapacak biçimde izleyen kesimde verilmiştir.

### Gereksiz Veri Tekrarı ve Veri Tutarsızlığının Önlenmesi

Önceki kesimde de belirtildiği gibi geleneksel dosya sistemlerinde her alt sistem ya da uygulama için gerekli olan veriler ayrı dosyalar biçiminde düzenlenmektedir. Böylece verilerden bir kısmı bilgisayarda aynı anda birden fazla dosya içinde yer alabilmekte dolayısıyla verinin büyük ölçüde tekrar edilmesi sonucuyla karşılaşmaktadır. Bu durum aynı zamanda veri derleme işleminin de tekrarlı olmasına ve veri derleme için yapılan harcamaların artmasına neden olabilmektedir. Diğer yandan veri tekrarı, verilerin depolanması için kullanılan bellek kapasitelerinin dolayısıyla donanım harcamalarının artmasına da yol açar. Veri tekrarı aynı zamanda tekrarlanan verilerin farklı dosyalardaki değerlerinin de farklı olabilmesine ve dolayısıyla veri tutarsızlığı ile uygulamalarda sorunlar yaşanmasına neden olabilir. Örneğin, bir işletmede yer alan ürünlere ilişkin *stok numarası* bazı dosyalarda *stok kodu* olarak nitelendirilmiş olabilir. Aynı işletmenin stoklar, satışlar ve giyim mağazalarına ilişkin veri dosyalarında aynı ürüne ilişkin ölçü birimi olarak bir dosyada XL kodunun bir başka dosyada 3 rakamı ile belirtilmesi tutarsızlığa bir başka örnektir.



Veritabanı yönetim sistemlerinde ise ilgili kurumda kullanılan bilgi sistemi bir bütün olarak düşünülmektedir. Buna göre farklı alt sistemlerin ihtiyaç duyacağı tüm veriler tek bir merkezde toplanarak ihtiyaç duyan her birimin istediği veriye bu merkezden ulaşması sağlanır. Böylece birden çok uygulamada kullanılan ortak verilerin tekrar edilmesinin önüne geçilir. Veritabanı yönetim sistemi kullanımıyla; veriler arası ilişkileri kurmak için T.C.Kimlik numarası, personel sicil numarası ve stok kodu gibi ayırteci nitelikte olan veri değelerinin zorunlu tekrarlanması dışında, veri tekrarı önlenir.

## Veri Bütünlüğünün Sağlanması

Veri bütünlüğü, veritabanında yer alan bir verinin farklı uygulamalarda kullanımı söz konusu olduğunda veri üzerinde yapılacak bir değışimin verinin kullanıldığı diğere uygulamalara da yansıtılması anlamına gelir. Bu, özellikle verinin silinmesi durumunda önem kazanmaktadır. Buna göre bir veri silindiğinde verinin ilişkili olduğu tüm uygulamalardan o verinin silinmesi gereklidir. Geleneksel dosya sistemlerinde veri bütünlüğünün sağlanması güç iken veritabanı yönetim sistemlerinde, sisteme girilen kısıtlamalar ve kurallar yardımıyla veri bütünlüğü rahatlıkla sağlanabilmektedir.

## Veri Paylaşımının Sağlanması

Geleneksel dosya sistemlerinde aynı veriye birden fazla kullanıcının aynı anda erişmesine olanak yoktur. Bu sistemlerde veriye erişim sıralı erişim kurallarına göre gerçekleştirilir. Buna göre veriye erişme isteğini ilk bildiren ilk sırada veriye erişir, diğere kullanıcının veriye erişebilmesi için ilk kullanıcının işinin bitmesi gerekir. Veritabanı yönetim sistemlerinde ise sunucu/istemci mimarisi kullanılarak veriler tek bir merkezden (sunucu bilgisayar), ağ üzerinde yer alan ve erişim yetkisi olan tüm kullanıcılara (istemci bilgisayarlar) aynı anda veriye erişme olanağı sunar. Bu durum verinin aynı anda birçok kullanıcı tarafından paylaşılması anlamına gelir. Verilerin ortak paylaşımı veritabanı yönetim sistemi yaklaşımının temel amaçlarından biridir.

Bununla birlikte uygulamaların veritabanında eş zamanlı işlem yapmasının doğuracağı sakıncaların da veritabanı tasarlanırken göz önüne alınması gerekir. Veritabanı yönetim sistemi yaklaşımında birtakım kısıtlama ve kurallarla söz konusu olumsuzluklar için önlem alınabilir.

Bununla ilgili olarak bankacılık sektöründen bir örnek verebiliriz. Bir bankanın veritabanı tanımları arasında “bir banka hesabından, hesaptaki mevcut miktardan daha fazla para çekilemez, hesap bakiyesi negatif olamaz” biçiminde ifade edilebilecek bir kısıtlamanın bulunduğunu varsayalım. Bu bankadaki bir hesapta 200.000 TL bulunduğunu ve bu hesaptan para çekmeye yetkili iki kişinin bulunduğunu düşünelim. Parayı çekmeye yetkili kişilerin her ikisi de aynı anda farklı şubelerden 100.000 ve 150.000 TL çekmeye çalışsın. Her iki işlem için hesaptaki mevcut miktara bakıldığında hesapta 200.000 TL bulunduğu ve bu miktarın çekilmek istenen miktardan büyük olduğu görülür. Eğer hiçbir önlem almadan her iki işlem de sürdürülürse hesap miktarı 250.000 TL azalarak hesap bakiyesi -50.000 TL ye ulaşır. Bu olumsuz durum veritabanı üzerinde tanımlanan bir kısıtlamayla önlenir. Örnekten de görüldüğü gibi veritabanı yönetim sistemi, verilerin aynı anda paylaşımına izin verirken, ortak paylaşımından doğabilecek olumsuz durumları da önleyebilecektir.

## Kullanımda Üst Düzey Uzmanlık Gerektirmemesi

Günümüzde kurumlar, bilgi sistemlerinden yoğun bir biçimde yararlanmakta ve bu bilgi sistemleri kapsamında birbiriyle ilişkili çok çeşitli ve büyük miktarda veriler üzerinde her türlü güncelleme, sorgulama, raporlama vb. işlemler yapmak istemektedir. Geleneksel dosya sistemlerinde veriler dosyalar biçiminde saklanırken kurumun ihtiyaç duyduğu



uygulamaları gerçekleştirmek üzere de ayrıca geliştirilen uygulama programlarından yararlanılmaktadır. Verilerin karmaşıklığı, veriler arasında çeşitli düzeylerde birçok ilişkilendirmenin yapılması gerekliliği ve uygulamaların çeşitliliği nedeniyle, kullanılan dosya yapıları, dosyalar arası ilişkileri kurmak için kullanılan teknikler ve dosyalara erişim için kullanılan yaklaşımlar oldukça karmaşık bir yapı sergilemektedir. Bu nedenle sözü edilen uygulama programlarının geliştirilmesi için bilişim teknolojisinde uzman kişilere ihtiyaç duyulmaktadır. Çeşitli verilere ulaşma ihtiyacında olan çok sayıda kullanıcı kesiminin bilgi sistemi ile etkileşimi de yalnızca sözkonusu bu uygulama programları ile sağlanabilmektedir. Bu durum sistemin kullanımını önemli derecede sınırlandırır. Çünkü uygulama programları yalnızca programın geliştirilmesi sırasında tanımlanmış olan genel ihtiyaçları karşılayabilir. Özellikle yönetici konumundaki kullanıcılar önceden belirlenmemiş çok seyrek ya da yalnız bir kez oluşan bilgi sistemi ihtiyaçlarını bu biçimde tanımlanmış uygulama programlarıyla karşılamakta zorlanırlar. Rutin olmayan bu tür ihtiyaçları karşılayacak programlar yine yalnızca bilişim teknolojisinde uzman olan kişilerce, uzun bir zaman içinde hazırlanabilir ve kullanıcıların bu kadar uzun süre beklemesi mümkün değildir. Bunun bir sonucu olarak da bu tür ihtiyaçlar zamanında karşılanamaz. Sonuç olarak bilgi sisteminin kullanımı sınırlı kalır. Bu noktada VTSY'lerin bu olumsuzlukları gideren iki yönü bulunmaktadır. Bunlardan birincisi, VTSY'lerin kullanıcıları verilerin saklanması için bilgisayar belleklerinde oluşturulan karmaşık fiziksel yapılarla ilgilenmek zorunda bırakmayıp, yalnızca ilgilendikleri verileri içeren basit mantıksal yapıları görmesine olanak sağlamasıdır. İkinci olarak da VTSY'lerin kullanıcılara veritabanı uygulamalarını kolaylıkla gerçekleştirebilmeleri için sorgu dili gibi kullanımı basit olanaklar sunmasıdır. Bunun sonucunda önceden tanımlanmamış veri ihtiyaçlarını karşılamak için bilişim uzmanına olan gereksinim ortadan kalkmış ve ilgili kullanıcılar, sorgu dillerini kullanmak yoluyla veri ihtiyaçlarını karşılayabilir duruma gelmiştir. Bu da kullanımın önceden belirlenmiş operasyonel uygulamalarla sınırlı kalmaması ve yaygınlaşması sonucunu getirmiştir.

## Verilerin Gizliliğinin ve Güvenliğinin Sağlanması

Herhangi bir kurumda, kurum çalışanları tarafından ortak kullanılan verilerin depolandığı yapının (dosya ya da veritabanı) ve bu yapı içinde yer alan verilerin gizliliği ve güvenliği çok önemli bir konudur. Veriler üzerinde okuma, yazma, ekleme, silme ve güncelleme gibi faaliyetler gerçekleştirilebilir. Ancak her veri üzerinde her kullanıcının bu faaliyetlerin tamamını gerçekleştirebilmesi, istenen bir durum değildir. Örneğin bir okulda öğrencilere ilişkin verilerle ilgilenen farklı kullanıcı grupları bulunur. Öğrencilerin kendileri, ders aldıkları öğretim üyeleri, idari personel, test biçimindeki sınav kâğıtlarının bilgi işlem teknolojileriyle okunmasını sağlayan teknik personel, sınav tarihleri ve sınav salonlarını planlayan personel, öğrenci işleri daire başkanlığı vb. bu kullanıcılara örnek olarak verilebilir. Bu kullanıcılardan bir kısmı veriler üzerinde yalnızca okuma işlemi gerçekleştirebilirken, bir kısmı hem okuma hem de yazma, diğer bir kısmı ise okuma, yazma ve güncelleme işlemlerini yapabilir. Örneğin öğrenciler sınavlardan aldıkları notları yalnızca görebilir (okuma faaliyeti) ama değiştiremezler (yazma faaliyeti). Öğretim üyesi öğrenci notlarını sisteme girebilir (yazma faaliyeti) ve onayladıktan sonra da görüntüleyebilir (okuma faaliyeti). Bununla birlikte onaylama işleminden sonra değişiklik yapamaz. Sisteme yanlış girilen bir notun düzeltilmesi ise bir dizi dilekçe ve alınan kararlar sonucu (öğrenci/öğretim üyesinin yazılı dilekçesi üzerine, yapılan incelemeler sonrasında alınan yönetim kurulu kararı ile) öğrenci işleri daire başkanlığındaki ilgili personel tarafından gerçekleştirilebilir. Öğrenci bilgi sistemi için örneklenen bu durum çeşitli kurumlardaki veriler için de örneklenebilir.

Verilere ve verilerin saklandığı yapılara erişim ve erişim sonrası çeşitli düzeylerde kul-lanma yetkisinin tanımlanması verilerin gizliliği ve güvenliğini sağlar. Bu anlamda gele-neksel dosya sistemlerinde gizlilik ve güvenliğin sağlanması pek kolay değildir.

Veritabanı yönetim sistemlerinde, veritabanı tasarlanır ve oluşturulurken veritaba-nı kullanıcılarının kimler olacağı ve hangi yetkilere sahip olacağı da belirlenir. Kullanıcı grupları ve yetkileri ile ilgili tanımlar veritabanındaki diğer tanımlarla birlikte saklanır. Herhangi bir kullanıcı veritabanı üzerinde işlem yapmak istediğinde, veritabanı öncelikle kullanıcının yapmak istediği işleme ilişkin yetkisinin olup olmadığını kontrol eder. Kul-lanıcı yetkisiz olduğu bir işlem yapmak istiyorsa veritabanı yönetim sistemi bu işlemin yapılmasına izin vermez. Veritabanı yönetim sistemlerinin bu özelliği verilerin yetkisiz kullanıcılar tarafından görülmesini engelleyerek verinin gizliliğini, verinin yetkisiz kulla-nıcılar tarafından değiştirilmesini önleyerek de verinin güvenliğini sağlamış olur.

## Standart Yapı ve Kuralların Uygulanabilir Olması

Geleneksel dosya sistemlerinde her alt sistem kendi faaliyetlerine özgü uygulamaları ve her uygulama da kendi dosya yapısını kullandığı için dosya yapılanmalarında belirli bir standart olmayıp farklılıklar bulunmaktadır.

VTSY'lerde ise merkezi bir kontrol sistemi bulunur. VTSY'lerin yapısal özellikleri ve veritabanı sorumlusunun varlığı veriler ve veritabanı üzerinde belirli standartların oluş-turulması ve uygulanabilmesini olanaklı kılar. Bu standartlar verinin yapısı, gösterim biçimi, adlandırılması, belgelenmesi ile ilgili yapısal standartlar olabileceği gibi kurum içi, kurumlar arası, ulusal ya da uluslararası düzeyde belirlenmiş kurallar biçiminde de olabilir. Standartların varlığı, veritabanı yapısını anlama ve kullanma açılarından büyük kolaylık sağlamanın yanı sıra farklı sistemler arasında veri alışverişi için de çok önemli bir ihtiyaçtır.

**Veritabanı yönetim sistemlerinin geleneksel dosya sistemlerine göre üstünlükleri nelerdir?**



SIRA SİZDE

## VERİTABANI KULLANICILARI

Veritabanı ile herhangi bir şekilde etkileşimde olan kişi ya da kişiler veritabanı kullanıcısı olup aşağıdaki gibi sınıflandırılabilirler:

- Veritabanı Sorumluları
  - Veritabanı Yöneticisi
  - Veritabanı Tasarımcısı
- Son Kullanıcılar
  - Standart Kullanıcılar
  - Sıradan ya da Parametrik Kullanıcılar
  - Gelişmiş Kullanıcılar
  - Bağımsız Kullanıcılar
- Sistem Analistleri ve Uygulama Programcıları

## Veritabanı Sorumluları

Veritabanı sorumluları, veritabanının tasarlanması, oluşturulması ve veritabanının işle-tim faaliyetlerinden birinci derecede sorumlu olan ve veritabanı üzerinde en fazla yetkiye sahip olan kullanıcılardır. Veritabanı sorumluları, *veritabanı yöneticisi* ve *veritabanı tasa-rımcısı* olarak iki başlık altında incelenebilir. Her iki sorumluluğu aynı kişi ya da kişiler alabileceği gibi veritabanını kullanacak olan kurumun/veritabanının büyüklüğü ve kul-lanıcı sayısı gibi faktörlere bağlı olarak veritabanı yöneticisi ile veritabanı tasarımcısının farklı kişiler olması da mümkündür.

### Veritabanı Yöneticisi

Veritabanı yöneticisinin (database administrator) veritabanına erişim yetkilerini belirleme, veritabanı kullanımının düzenlenmesi ve izlenmesini sağlama, ihtiyaç duyulan yazılım ve donanım kaynaklarını edinme biçiminde sıralanan sorumlulukları vardır. Ayrıca güvenlik ihlalleri ve kötü sistem yanıt süresi gibi sorunların çözümünden de sorumludur. Büyük işletmelerde bu sorumluluklar için yardımcı personele de ihtiyaç duyulur.

### Veritabanı Tasarımcısı

Veritabanı tasarımcısı (database designer) veritabanında saklanacak olan verilerin tanımlanmasından ve bu verilerin depolanması ve gösterilmesi için gerekli olan uygun yapının seçilmesinden sorumludur. Bu görevler çoğunlukla verilerin veritabanına depolanmasından ve veritabanı uygulamalarından önce yerine getirilir. Veritabanı tasarımcısı, muhtemel veritabanı kullanıcılarının ihtiyaçlarını anlamak ve onların bu ihtiyaçlarını karşılayabilecek özellikte bir tasarımı oluşturmak amacıyla öncelikle sözkonusu veritabanı kullanıcıları ile iletişime geçmelidir. Pek çok uygulamada, veritabanı tasarımcısı veritabanı yöneticisinin yardımcı personeli olup veritabanı tasarımı tamamlandıktan sonra bu personele başka sorumluluklar atanmaktadır. Veritabanı tasarımcısı tasarım boyunca potansiyel kullanıcı gruplarıyla karşılıklı etkileşim hâlinde olup bu grupların veriye erişimini ve veri üzerinde işlem yapabilmelerini olanaklı kılan kullanıcı görünümünü geliştirirler. Her kullanıcı grubuna ilişkin görünüm, analiz edilerek diğer kullanıcı gruplarına ilişkin görünümle bütünleştirilmeleri sağlanır. Veritabanı tasarımının tüm kullanıcı gruplarının ihtiyaçlarını destekleyecek kapasitede olması gereklidir.

Veritabanı yöneticisi ya da veritabanı tasarımcısı biçiminde ayırım yapmaksızın, veritabanı sorumlularının yerine getirdikleri temel görevler aşağıdaki gibi özetlenebilir:

- *Veritabanı tasarımını yapma:* Veritabanının farklı düzey şemalarının oluşturulması, veriler üzerinde yapılacak her türlü işlem için gereksinimlerin belirlenmesi, veritabanı içeriğinin oluşturulması.
- *Bütünlük kısıtlamalarını belirleyip tanımlama:* Veritabanında veri bütünlüğünün sağlanabilmesi, veri kaybının önüne geçilebilmesi, veri bütünlüğünü tehlikeye sokacak kullanıcı hatalarının önlenmesi amacıyla gerekli kurallar, ilişkiler ve kısıtlamaların belirlenmesi.
- *Veritabanı kullanım yetkilerini tanımlama:* Veritabanı kullanıcılarının ve kullanım yetkilerinin tanımlanması ile her kullanıcı grubunun hangi veriler üzerinde hangi işlemleri yapmaya yetkili olduğunun belirlenmesi. Bu tanımlamalar sonrasında ilgili kullanıcılara yetkili oldukları erişimin sağlanması.
- *Veritabanı güvenliğini sağlama:* Kullanıcıdan ya da yazılım sorunlarından kaynaklanabilecek veri kaybının önlenmesi amacıyla veritabanının yedeklerinin alınması ve kurtarma işlemlerine ilişkin düzenlemelerin yapılması.
- *Veritabanının işletimini izleme ve sürekliliğini sağlama:* Veritabanının kullanıma sürekli açık olmasının sağlanması ve sistemde meydana gelebilecek herhangi bir sorunu hızlı biçimde giderebilecek tedbirlerin alınması.
- *Güncelleme ihtiyaçlarına cevap verebilme:* Kullanıcı ihtiyaçlarında ortaya çıkabilecek değişikliklerin izlenebilmesi ve bu değişikliklere paralel olarak veritabanı içeriği, şema tanımları, bütünlük kısıtlamaları, fiziksel yapı ile ilgili parametreler, kullanıcı tanımları ve kullanıcı yetkilerinde gerekli değişikliklerin oluşturulması ve tanımlanması.
- *Veritabanından beklenen performansı sağlama:* Veritabanı kullanıcılarının beklentilerine cevap verecek bir yapının donanım ve yazılım ihtiyaçlarının sağlanması, veritabanındaki tanımlamaların, kısıtlamaların kullanıcı ihtiyaçlarına cevap verebilecek yeterlilikte olup olmadığının izlenmesi, gerekiyorsa değişikliklerin yapılması.

## Son Kullanıcılar

Son kullanıcılar (end users), yaptıkları işler gereği veritabanına sorgulama ya da güncelleme yapmak veya rapor üretmek için erişen kullanıcılardır. Bu tür kullanıcılar veritabanı ile kullanıcının bağlantısını sağlayan ve uygulama programcıları tarafından geliştirilen yazılımları kullanırlar. Son kullanıcılar da kendi içinde gruplandırılabilir:

**Standart son kullanıcılar:** Veritabanına nadiren erişim yapan fakat her seferinde farklı bilgi ihtiyacı olabilen kullanıcılardır. Bu tür kullanıcılar isteklerini belirtmek için gelişmiş veritabanı sorgu dili kullanırlar. Orta ya da üst düzey yöneticiler bu gruba örnek verilebilir.

**Sıradan ya da parametrik son kullanıcılar:** Son kullanıcıların önemli bir bölümünü bu tür kullanıcılar oluşturur. Bu kullanıcıların temel iş fonksiyonları, veritabanı üzerinde sürekli bir sorgulama ve güncelleme yapmalarını gerektirir. Standart (önceden belirlenmiş) sorgu ve güncelleme yaparlar. Bu gruba giren kullanıcılar çok çeşitli konum ve görevlerde olabilirler. Örneğin;

- Banka memurunun, mevduat hesaplarına yatırılan ya da çekilen miktarları veya hesap bakiyelerini kontrol etmesi
- Havaalanları ve oteller için rezervasyon yapan acentaların ya da araç kiralama şirketlerinin, gelen müşteri taleplerini cevaplayabilmek amacıyla sistemlerinde uygun yer ya da araç olup olmadığını sorgulamaları ve buna göre rezervasyon yapmaları

sıradan son kullanıcılara ve veritabanını kullanma biçimlerine verilebilecek örnekler arasındadır.

**Gelişmiş son kullanıcılar:** VTYS'nin sağladığı özellikler yardımıyla ayrıntılı olarak belirledikleri karmaşık gereksinimlerini karşılamak amacıyla veritabanını kullanan gruptur. Mühendisler, bilim adamları, işletme analistleri vb. bu gruptaki kullanıcılara örnek olarak verilebilir.

**Bağımsız son kullanıcılar:** Bu kullanıcılar menü kullanımı ya da araç çubukları gibi grafiksel öğeler yardımıyla kullanım kolaylığı sağlayan hazır paket programlarını kullanarak kişisel veritabanlarının sürekliliğini sağlar. Vergilere ilişkin olarak hazırlanan ve kişisel finansal verileri depolayan vergi paketi kullanıcıları bu tür kullanıcılara verilebilecek bir örnektir.

Tipik bir VTYS, veritabanına erişim için farklı kullanıcı gruplarının ihtiyaç duyduğu birçok özelliği sağlar. Her kullanıcı grubu bu özelliklerden ihtiyaç duyduğu kadarıyla yararlanır. Örneğin, sıradan son kullanıcıların VTYS tarafından sağlanan özellikler hakkında bilgi sahibi olma ihtiyacı çok az olacaktır. Bu tür kullanıcıların, veritabanına erişimi sağlayan kullanıcı ara yüzlerini kullanmayı ve standart işlemlere ilişkin uygulamaları bilmeleri yeterli olacaktır. Standart son kullanıcılar da VTYS özellikleri içinden yaptıkları işin gereği olarak, yalnızca sürekli tekrarlayacakları özellikleri öğrenirler. Gelişmiş son kullanıcılar ise karmaşık gereksinimlerini karşılayabilmek için VTYS'nin pek çok özelliğini öğrenmeye çalışırlar. Bağımsız son kullanıcılar da veritabanı ihtiyaçlarını karşılamak üzere belirli bir yazılım paketini kullanma konusunda ustalaşırlar.

## Sistem Analistleri ve Uygulama Programcıları

Sistem analisti son kullanıcıların, özellikle de sıradan son kullanıcıların gereksinimlerini belirleyen ve standart işlemler yoluyla bu gereksinimleri karşılayabilecek ayrıntıları belirleyen kişi ya da kişilerdir. Uygulama programcıları ise sistem analisti tarafından belirlenen ayrıntıları program hâline getiren ve daha sonra test eden, hataları ayıklayan, belgeleyen ve kaydedilmiş işlemler olarak sürekliliğini sağlayan kişilerdir. Yaygın olarak yazılım geliştiriciler ya da yazılım mühendisleri olarak da anılan analistler ve programcıların yukarıda sıralanan görevlerini yerine getirebilmeleri için VTYS'nin sağladığı tüm özellikleri bilmeleri gerekir.



## VERİTABANI YÖNETİM SİSTEMLERİNİN MİMARİSİ

VTYS'lerinin mimarisi geçmişten günümüze incelendiğinde, ilk veritabanı sistemlerinde, VTYS'nin tüm yazılım paketlerinin tek bir sisteme entegre edildiği, modern VTYS'lerinde ise istemci/sunucu mimarisi ile modüler bir yapılanmanın sözkonusu olduğu görülmektedir. Buna bağlı olarak, büyük merkezi ana bilgisayarlar yerini, iletişim ağları aracılığıyla çeşitli sunucu bilgisayarlara bağlanan yüzlerce iş istasyonu ve kişisel bilgisayarlara bırakmıştır. Burada sözü edilen sunucu bilgisayarlar; web sunucular, veritabanı sunucuları, dosya sunucuları, uygulama sunucuları vb. olarak örneklenebilir.

Basit bir istemci/sunucu VTYS mimarisinde sistem fonksiyonel olarak iki modüle ayrılmaktadır. İstemci modülü (client modül), VTYS'nin herhangi bir kullanıcı iş istasyonunda ya da kişisel bilgisayar üzerinde çalışan parçasıdır. Tipik olarak istemci modülünde veritabanına erişmek için uygulama programları ve kullanıcı arayüzleri bulunur. Sunucu modülünde (server modüle) ise veri deposu ile veri deposuna erişimi ve sorgulamayı sağlayacak fonksiyonlar yer alır.

## Veri Modelleri

Veritabanı yaklaşımının temel karakteristiklerinden biri, veritabanının bazı veri soyutlama düzeyleri sağlamasıdır. **Veri soyutlama** (data abstraction), verilerin düzenlenmesi ve depolanmasına ilişkin ayrıntıların gizlenmesi ve verinin daha iyi anlaşılmasını sağlamak için veriye ilişkin temel özelliklerin vurgulanması anlamına gelir. Veri modeli ise söz konusu bu soyutlamaları gerçekleştirebilmek için gerekli olan araçları sağlar.

**Veri modeli** (data model), bir veritabanının mantıksal yapısını tanımlamada kullanılacak kavramlar, işlemler ve kurallar bütünüdür. Veritabanının mantıksal yapısı; veri tipleri, veriler arasındaki ilişkiler, veri üzerinde uygulanacak kısıtlamalar vb. dir. Veri modellerinin çoğu verinin geri çağırılması ve veritabanı üzerinde güncelleme yapmaya yönelik işlemleri de içerir. Veri modelinin bu temel işlevlerine ek olarak veritabanının dinamik hareketlerini belirleyen kavramları içermesi de yaygınlaşmıştır. Bu özellik veritabanı tasarımcısına veritabanı üzerinde, geçerli kullanıcı tanımlı işlemleri yapmasına olanak sağlar.

## Veri Modellerinin Sınıflandırılması

Yüksek düzeyli ya da kavramsal veri modelleri (conceptual data models), kullanıcıların *veri algılama biçimiyle* ilişkili kavramları kapsar.

Düşük düzeyli ya da **fiziksel veri modelleri** (physical data models), *verinin bilgisayar ortamında nasıl depolanacağına* ilişkin ayrıntıları tanımlayan kavramları kapsar. Fiziksel veri modelleri kavramları genellikle son kullanıcılar için değil, bilgisayar uzmanları için geliştirilir.

Bu iki uç model arasında kalan model sınıfı ise *temsili modeller* ya da *uygulama veri modelleri* (implementation data model) olarak adlandırılır. Bu model hem son kullanıcılar tarafından kolaylıkla anlaşılabilen kavramları hem de verinin bilgisayarda depolanması yöntemlerine ilişkin kavramları içerir.

Kavramsal veri modelleri; varlıklar, öznitelikler, ilişkiler gibi kavramları kullanır. Bu kapsamda varlık-ilişki modelleme konusu 2.Ünite'de ele alınmıştır. Uygulama veri modelleri geleneksel ticari VTYS'lerinde en sık kullanılan modellerdir. Bu modeller geçmişte yaygın olarak kullanılan ağ ve hiyerarşik veri modeli ile günümüzde yaygın olarak kullanılan ilişkisel modelleri içerir. Uygulama veri modelleri, verileri kayıt yapılarıyla gösterdiklerinden bazen kayıt tabanlı veri modelleri olarak da adlandırılırlar.

Veritabanı yaklaşımı **veri soyutlama** özelliği ile farklı kullanıcıların verileri tercih ettikleri ayrıntı seviyesinde algılamalarına, gereksiz ayrıntıları görmemelerine olanak sağlar.

Temel **veri modeli** işlemleri; veritabanı üzerinde ekleme, silme, değiştirme, veriyi geri çağırma gibi genel işlemleri içerir.

**Fiziksel veri modelleri**, kayıt biçimi, kayıt sırası ve erişim yolu bilgilerini göstermek suretiyle verilerin bilgisayarda dosya olarak nasıl saklandığını belirler.

Nesneye yönelik modeller kavramsal veri modeliyle yakın ilişkili olan yüksek düzeyli uygulama modellerinin yeni üyesi olarak düşünülebilirler. Nesneye yönelik veri modelleri için bir standart ODGM (Object Data Management Group-Nesneye Yönelik Veri Yönetim Grubu) olarak adlandırılır. Nesneye yönelik veri modelleri aynı zamanda (özellikle yazılım mühendisliği alanında) yüksek seviyeli kavramsal model olarak da kullanılmaktadır.

## Şemalar, Örnekler ve Veritabanının Durumu

Veritabanının herhangi bir veri modeliyle tanımlanması *veritabanı şeması* (database schema) olarak adlandırılır. Sözkonusu bu şema veritabanının tasarlanması sürecinde oluşturulur ve sık sık değişiklik göstermez. Veri modellerinin çoğunda şemaların bir diyagram olarak gösterilebilmesi amacıyla kullanılan belirli kuralları vardır. Şema görünümleri *şema diyagramı* (schema diagram) olarak adlandırılır. Şema diyagramları şemaların yalnızca bazı yönlerini gösterir. Diğer yönler diyagramdan belirlenemez. Örneğin, her bir veri kalemine ilişkin veri tipi, çeşitli dosyalar arasındaki ilişkiler ya da verilere ilişkin kısıtlar diyagramından izlenemez. Veritabanındaki mevcut veriler oldukça sık değişebilir. Örneğin, bir öğrenci veritabanında her yeni öğrenci kaydında veritabanı değişecektir. Bir veritabanının herhangi bir andaki durumuna *anlık görüntü* denir. Bu aynı zamanda veritabanının mevcut *örnek* (instance) kümesi olarak da adlandırılır. Veritabanında her şema yapısı kendi oluşum kümesine sahiptir. Örneğin “öğrenci” veritabanındaki bireysel öğrenci kayıtları oluşumlara örnekler. Veritabanı durumu, belirli bir şema ile ilişkilendirilerek yapılandırılır. Her zaman yeni bir kayıt ekleme, silme ya da bir kayıttaki verinin değerini değiştirmek mümkündür (böylece veritabanının durumu bir durumdan diğerine değiştirilebilir).

VTYS şema yapılarını ve kısıtlarını (meta veri) veritabanı içinde tanımlar ve VTYS yazılımı ne zaman ihtiyaç duysa bu şemaya başvurur.

## Üç Şema Mimarisi

Veritabanı yaklaşımının önemli karakteristikleri aşağıdaki biçimde sıralanabilir:

1. Veritabanı tanımlarını (şema) depolamak için kendi kendine tanım yapabilmeyi sağlayan bir katalog kullanır.
2. Veri ve program izolasyonunu sağlar (program – veri bağımsızlığı; program – işlem bağımsızlığı)
3. Çoklu kullanıcıyı destekler

Üç şema mimarisi, yukarıda sıralanan bu özelliklerin gerçekleştirilmesine ve anlaşılmasına yardımcı olacaktır.

Üç şema mimarisinin amacı kullanıcı uygulamalarını fiziksel veritabanından ayırt etmektir. Bu mimaride şemalar izleyen üç düzeyde tanımlanır:

1. **İçsel (fiziksel) düzey** (internal level), veritabanının *fiziksel depolama yapısını* tanımlayan içsel şemayı içerir. İçsel şema, veriyi depolama ayrıntılarının tamamını ve veritabanına erişim yollarını tanımlayan fiziksel veri modelini kullanır.
2. **Kavramsal düzey** (conceptual level), kullanıcı topluluğu için tüm veritabanının yapısını tanımlayan kavramsal şemayı içerir. Kavramsal şema fiziksel depolama yapısının ayrıntılarını gizler ve veritabanında yer alan verilerin tipine, veriler arası ilişkilere, kullanıcı işlemlerine ve kısıtlara ilişkin tanımlara yoğunlaşır. Veritabanı sistemlerinde, uygulama veri modelleri kavramsal şemayı tanımlamak için kullanılır. Bu kavramsal şema, yüksek düzeyli kavramsal veri modelleri kullanılarak tasarlanır.
3. **Dışsal (görünüm) düzey** (external level), bir dizi dışsal şema ya da kullanıcı görünümü içerir. Her dışsal şema bir grup kullanıcının ilgilendiği bazı veritabanı bölümlerini tanımlar. Böylece veritabanının diğer kısmı bu kullanıcı grubundan gizlenir. Her dışsal şema, bir yüksek düzey veri modelinde tasarlanan dışsal şema tabanlı uygulama veri modeli kullanır.



Çoğu VTYS tam ve açık olarak bu üç düzeye ayrılmaz fakat bu üçlü şema mimarisini bir ölçüde destekler. Örneğin bazı eski VTYS'leri fiziksel düzey ayrıntılarını ayrı bir düzey olarak ele almayıp kavramsal şema içinde barındırmaktaydı.

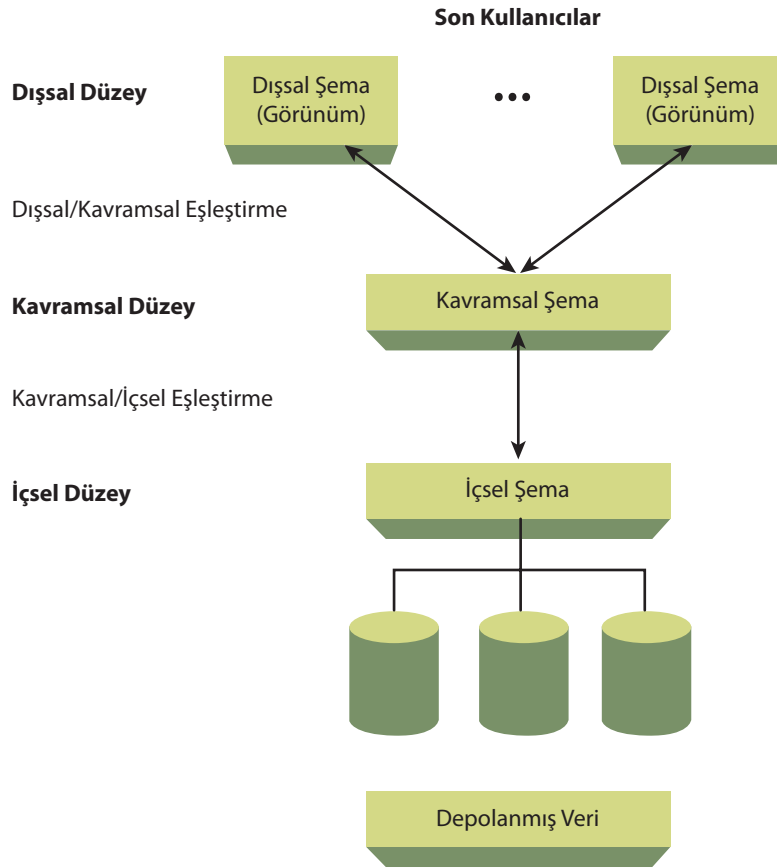
Burada dikkat edilmesi gereken nokta, bu üç şemanın verilerin yalnızca tanımı olduğudur. Depolanmış gerçek veri yalnızca içsel (fiziksel) düzeydedir.

Şekil 1.2'den de izlenebileceği gibi üç şema mimarisine dayanan VTYS'lerinde her kullanıcı grubu kendi dışsal (görünüm) şemalarına başvurur. Bu nedenle VTYS, dış şemada yapılan belirli bir isteği kavramsal şema isteğine daha sonra da depolanmış veritabanı üzerinde işlem yapmak üzere içsel şema isteğine dönüştürmek zorundadır. Yapılan istek veritabanından bilgi çağırarak ise veritabanından çıkarılan veri, kullanıcının dışsal görünümüyle eşleştirecek biçime dönüştürülür. Şema düzeyleri arasında bu şekilde gerçekleştirilen dönüşümler *eşleştirme* (mapping) olarak adlandırılır. Bu şekildeki eşleştirmeler zaman alıcı olabilir bu nedenle bazı VTYS'leri dışsal görünümünü desteklemezler. Buna rağmen bu sistemlerde bile kavramsal ve içsel düzeyler arasında belirli miktarda dönüşüm gerekli olacaktır.

**Şekil 1.2**

Üç Şema Mimarisi

**Kaynak:** Elmasri, R. ve Navathe S.B., (2011, s.34)



## Veri Bağımsızlığı

**Veri bağımsızlığı**, herhangi bir düzeydeki şema değiştiğinde bir üst düzeydeki şemanın değişmeden kalmasını bununla birlikte iki düzey arasındaki eşleştirmelerin değişmesini sağlar. Bu eşleştirme sonucunda bir üst düzeydeki şemaya başvuran uygulama programları değişiklik ihtiyacı göstermez.

Şema düzeyleri arasındaki dönüştürme kapasitesi olarak tanımlanabilen veri bağımsızlığı (data independence) yine üç şema mimarisi ile açıklanabilir. Veri bağımsızlığı iki başlıkta ele alınır:

1. **Mantıksal veri bağımsızlığı** (logical data independence), kavramsal şemanın dışsal şemalarda ya da uygulama programlarında değişiklik yapılmaksızın değiştirilebilmesi anlamına gelir. Kavramsal şemada değişiklik, veritabanını genişletmek (bir kayıt tipi ya da veri kalemi eklemek) kısıtları değiştirmek ya da veritabanını eksiltmek (bir kayıt ya da veriyi silmek) biçiminde olabilir.



2. **Fiziksel veri bağımsızlığı** (physical data independence), kavramsal şemada bir değişiklik yapılmaksızın içsel şemada değişiklik yapma kapasitesidir. Buna bağlı olarak içsel şemada yapılan değişiklik dışsal şemalarda da bir değişiklik yapılmasını gerektirmez. İçsel şemalarda değişiklik ihtiyacı fiziksel dosyalar üzerinde yeniden düzenleme ihtiyacı ile ortaya çıkabilir. Veritabanına ek erişim yapısının oluşturulması, veritabanının bilgiyi çağırma performansının geliştirilmesi ya da veritabanının güncellenmesi içsel şema değişikliklerine verilebilecek örneklerdir.

Fiziksel veri bağımsızlığı, fiziksel ayrıntıların olduğu çoğu veritabanı ve dosya ortamında mevcuttur. Bununla birlikte kavramsal veri bağımsızlığına erişmek daha zordur çünkü kavramsal veri bağımsızlığı uygulama programlarını etkilemeden yapısal değişiklikler ve kısıtlama değişikliklerine izin verir. Çok düzeyli VTSY'leri söz konusu olduğunda bu VTSY'lerin, farklı düzeyler arasındaki veri ve istek eşleştirmelerini haritalayan bir kataloğu da (fihrist) barındırması gerekir. VTSY bu amaçla ek bir yazılım kullanır.

## Veritabanı Yönetim Sistemlerinde Kullanılan Diller

VTYS daha önce de belirtildiği gibi farklı özellikteki kullanıcılara destek verir. Bu nedenle VTYS'lerin bu farklı kullanıcı gruplarından her birine yönelik uygun dil ve arayüz kullanmaları gerekir.

Veritabanı tasarımı tamamlandıktan ve veritabanını uygulayacak bir VTYS seçildikten sonra ilk adım veritabanı için kavramsal ve fiziksel şemaların belirlenmesi ve bu iki düzey arasında eşleştirmelerin yapılmasıdır. Düzeyler arasında mutlak bir ayrımın yapılmadığı pek çok VTYS'de iki şemayı tanımlamak amacıyla **veri tanımlama dili** (data definition language) olarak adlandırılan bir dil, veritabanı yöneticisi ve veritabanı tasarımcısı tarafından kullanılır. İki şema bu şekilde tanımlandıktan sonra sözkonusu tanımlar, VTYS'de **veri tanımlama dili derleyicisi** (data definition language compiler) tarafından işlenerek VTYS kataloğunda depolanacak uygun yapılar biçimine dönüştürülür.

VTYS veritabanı tanımlarının, VTYS tarafından derlenerek saklanması veritabanı yaklaşımının temel özelliklerinden biridir. Bu süreç sayesinde, veritabanı tanımlarının yetkili kişiler tarafından bir kez yapılması, tanımların kalıcılığının sağlanması ve kullanıcıların bu tanımları kullanmaları ve bu tanımlara uygun işlem yapmaları sağlanmış olur (Yarımagan, 2000, s.2).

Veritabanında veri tanımlarının yer aldığı yapı **veri sözlüğü** (data dictionary) olarak da adlandırılabilir.

Kavramsal ve fiziksel düzeyler arasında açık biçimde ayrımın olduğu VTYS'lerde veri tanımlama dili yalnızca kavramsal şemayı belirlemek için kullanılır. Böylesi VTYS'lerde fiziksel şemayı belirlemek içinse **depolama tanımlama dili** (storage definition language) adı verilen bir başka dilden yararlanılır. İki şema arasındaki eşleştirme ise veri tanımlama dili ya da depolama dilinden biri kullanılarak belirlenir.

Günümüzde kullanılan ilişkisel VTYS'lerin çoğunda depolama tanımlama dilinin görevini yerine getiren belirli bir dil yoktur. Bunun yerine fiziksel şema fonksiyonları, parametreler ve depolamayla ilgili ayrıntıların birleşimiyle belirlenir. Bu, veritabanı yönetici personeline indeksleme seçenekleri ve depolama için verinin eşleştirilmesini kontrol etme izni verir.

Üç şemalı mimarinin tam anlamıyla kullanıldığı yapılarda **görünüm tanımlama dili** (view definition language) olarak adlandırılan üçüncü bir dile ihtiyaç duyulur. Görünüm tanımlama dili, kullanıcı görünümünü belirlemek ve bunların kavramsal şemadaki eşleştirmelerini belirlemek için kullanılır. Fakat pek çok veritabanının **veri tanımlama dili** hem kavramsal hem de dışsal (görünüm) şemaların her ikisini de tanımlamak için kullanılır. Örneğin, ilişkisel VTYS'lerinde SQL (yapısal sorgulama dili) görünüm tanımlama dili rolüyle tanımlanır ve önceden tanımlanmış bir sorgunun sonucu olarak kullanıcı ya da uygulama görünümünü tanımlar.

Veritabanı tanımları **Veri Tanımlama Dili** kullanılarak oluşturulur. Daha sonra bu tanımlar **Veri Tanımlama Dili Derleyicisi** kullanılarak çözümlenir ve gerektiğinde kullanıcıya iletilmek üzere VTYS tarafından uygun yapılara dönüştürülerek saklanır.

Veritabanı şemaları yukarıda sözü edilen biçimde tanımlanıp derlendiğinde ve ilgili veriler veritabanına kaydedildiğinde veritabanı, üzerinde veri düzenleme işlemi yapılabilir duruma gelir. Burada sözü edilen veri işleme; veritabanından veri çağırma, veri ekleme, veri silme ve veri üzerinde değişiklikler yapma faaliyetlerini içerir. VTYS bu işleme faaliyetlerini gerçekleştirebilmek amacıyla *veri işleme dili* (data manipulation language) olarak adlandırılan bir dil kullanır. Veritabanından bilgi alma amacıyla sorgulama yapmak için kullanılan veritabanı dili ise *sorgulama dili* (query language) olarak adlandırılır.

Günümüzde kullanılan VTYS'lerinde yukarıda belirtilen dil çeşitleri genellikle birbirinden ayrı diller olarak düşünülmez aksine tüm dil çeşitlerinin görevini yerine getiren geniş kapsamlı birleştirilmiş bir dil kullanılır. Kapsamlı birleştirilmiş dile tipik örnek ilişkisel veritabanı dili *SQL (Structured Query Language-Yapısal Sorgulama Dili)*'dir.

### Veritabanı Yönetim Sistemlerinin Bileşen Modülleri

Veritabanı ve VTYS kataloğu genellikle disk üzerinde depolanır. Diske erişim, öncelikle disk üzerinde okuma/yazma işlemlerini programlayan işletim sistemi tarafından kontrol edilir. Çoğu VTYS'nin, diskin okunup/yazılmasını programlamak için kendi *ara bellek yönetim modülü* (buffer management module) bulunur. Böylece diskin okuma/yazma görevini azaltan bu modül veritabanı performansını dikkate değer ölçüde arttırır. VTYS'nin daha yüksek düzey modülü olan *depolanmış veri yöneticisi* (stored data manager) ise veritabanının ya da kataloğun bir bölümü olsun ya da olmasın, diske depolanmış VTYS bilgilerine erişimi kontrol eder.

Şekil 1.3, tipik VTYS bileşenlerini basitçe açıklamaktadır. Şekil iki parçaya ayrılmış olup üst kısmı veritabanının çeşitli kullanıcılarını ve veritabanı ile etkileşimlerini göstermektedir. Alt kısmı ise, verilerin depolanması ve veri üzerinde işlem yapmaktan sorumlu iç bileşenlerini göstermektedir.

Öncelikle Şekil 1.3'ün üst kısmını inceleyelim. Bu kısım, veritabanı yönetici personelleri, sorguları formüle etmek için etkileşimli ara yüzlerle çalışan standart kullanıcılar, programlama dillerini kullanarak program oluşturan uygulama programcıları ve önceden tanımlanmış işlemler için parametreler sağlayarak veri giriş işlerini yapan sıradan kullanıcılar için ara yüzleri göstermektedir. Veritabanı yönetici personelleri veri tanımlama dili ve veritabanı yönetici personeline özgü komutlar yardımıyla veritabanı tanımlarını ve veritabanı üzerinde gerekli değişiklikleri yaparlar.

Veri tanımlama dili derleyicisi, veri tanımlama dilinde belirlenmiş şema tanımlarını işler ve şemaların tanımlarını (meta-veri) VTYS kataloğuna depolar. Katalog; dosyaların isimleri ve boyutları, veri kalemlerinin adları ve veri türleri, her dosyanın depolama ayrıntıları, şemalar arasındaki bilgi eşleştirmesi ve kısıtlamalar gibi bilgileri içerir. Ayrıca katalog, VTYS modüllerinin ihtiyaç duyduğu birçok bilgi türünü de depolar. Böylece daha sonra ihtiyaç duyulan bilgiler için kataloğa başvurulabilir.

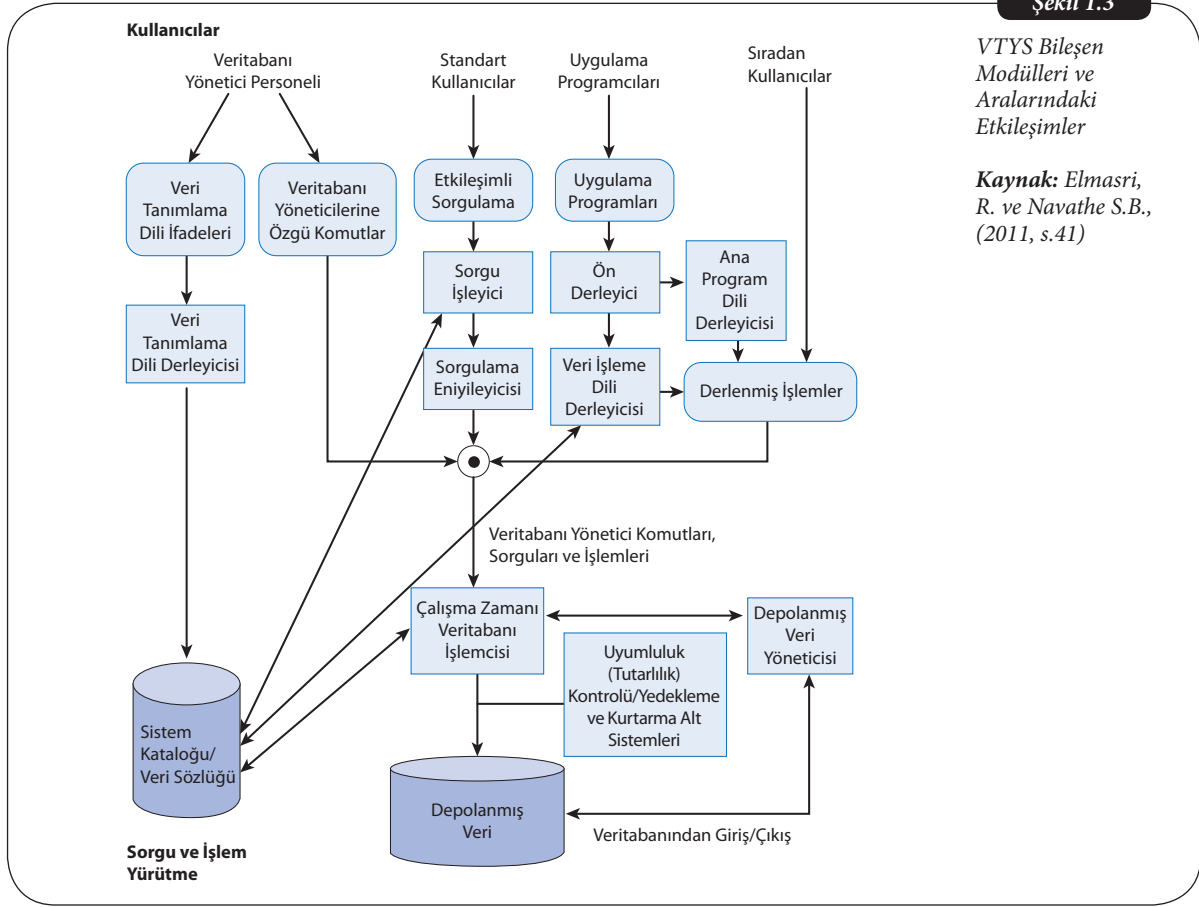
Veritabanından nadiren bilgi ihtiyacı olan standart kullanıcılar, *etkileşimli sorgulama* olarak adlandırılan, bir tür ara yüz formu kullanarak veritabanı ile etkileşime geçerler.

Veritabanına iletilen sorguların, anlamsal ve sözdizimsel çözümlemesi, kullanıcı yetkisinin kontrolü ve sorguları işlemek için hangi algoritmanın kullanılması gerektiğini belirleme işi *sorgu işleyici* (query processor) adı verilen bileşen tarafından gerçekleştirilir. Sorgu işleyicinin alt bileşenlerinden biri olan *sorgu eniyileyici* (query optimizer) ile işlemlerin yeniden düzenlenmesi ve yeniden sıralanması, fazlalıkların elenmesi ve yürütüm boyunca doğru algoritma ve indekslerin kullanılmasıyla sorgunun en iyi biçimde işletilmesi sağlanır. Sorgu eniyileyici, depolanmış veri hakkında istatistiki ve diğer fiziksel bilgiler için sistem kataloğuna başvurarak sorgu için gerekli işlemleri uygulayan yürütülebilir kodları üretir. Söz konusu bileşen bu işlemleri yaparken ayrıca çalışma zamanı işlemcisiyle de etkileşim hâlinindedir.

Şekil 1.3

VTYS Bileşen Modülleri ve Aralarındaki Etkileşimler

**Kaynak:** Elmasri, R. ve Navathe S.B., (2011, s.41)



Uygulama programcıları, Java, C veya C++ gibi ana programlama dillerinde yazılan ve bir ön derleyiciye gönderilen programlar yazar. Ön derleyici ana programlama dilleri ile yazılan ve uygulama programından gelen veri işleme komutlarını çözümler. Programın geri kalan komutları da ana programlama dili derleyicisine gönderilir. Veri işleme komutları ve programın geriye kalan komutları; yürütüm kodları, çalışma zamanı veritabanı işlemcisi ile bağlantılı kılınarak kaydedilir. Kaydedilmiş işlemler, parametrelerle işlem yapan sıradan kullanıcılar tarafından tekrar tekrar yürütülür. Her yürütüm, ayrı bir işlem olarak düşünülür. Buna örnek olarak, hesap numarası ve para miktarı parametreleri kullanılarak bankadan para çekme işlemi verilebilir.

Şekil 1.3'ün alt kısmında, *çalışma zamanı veritabanı işlemcisi* (runtime database processor) veritabanı yöneticilerine özgü komutları, yürütülebilir sorgu planlarını ve parametreleri önceden belirlenmiş işlemleri yürütür. Sistem kataloğu ve veri sözlüğü ile birlikte çalışır ve onları güncelleyebilir. Ayrıca, disk ve ana bellek arasındaki düşük düzey girdi/çıkış (okuma/yazma) işlemlerini yerine getirmek amacıyla temel işletim sistemi hizmetlerini kullanan *depolanmış veri yöneticisi* ile de etkileşim hâlinindedir. Şekilde, *uyumluluk kontrolü ile yedekleme ve kurtarma alt sistemleri* tek modül olarak gösterilmiştir. Bunlar, veritabanı işlemlerinin yönetimi için çalışma zamanı veritabanı işlemcisiyle bütünleştirilmiştir.

*Veri sözlüğü* şemalar ve kısıtlar hakkında kataloğ bilgisini depolamanın yanı sıra tasarım kararları, kullanım standartları, uygulama programı açıklamaları ve kullanıcı bilgileri gibi diğer bilgileri de depolar. Böyle bir sistem *bilgi havuzu* olarak da adlandırılır. Bu bilgiye kullanıcılar veya veritabanı yönetici personeli tarafından gereksinim duyulduğunda doğrudan erişilebilir. Veri sözlüğü VTYS kataloğuna benzemekle birlikte daha geniş bilgi çeşitleri içerir ve esas olarak VTYS yazılımından ziyade kullanıcılar tarafından erişilir.

## VERİTABANI TÜRLERİ

Daha önce de söz edildiği üzere her veritabanı yönetim sistemi bir veri modeli kullanır. Veritabanında yer alacak veriler ve veriler arasında kurulacak ilişkiler mantıksal olarak ilgili veri modeline göre yapılandırılır ve veritabanları da buna göre sınıflandırılır.

Geçmişten günümüze kadar geliştirilmiş olan çok sayıda veri modeli, kullandıkları teknikler açısından dört temel başlıkta incelenir. Bu dört veri modelinden hangisini kullandığına bağlı olarak veritabanları da aşağıda verilen dört başlık altında sınıflandırılabilir:

1. Hiyerarşik veritabanı (Hierarchical database)
2. Ağ veritabanı (Network database)
3. İlişkisel veritabanı (Relational database)
4. Nesneye yönelik veritabanı (Object oriented database)

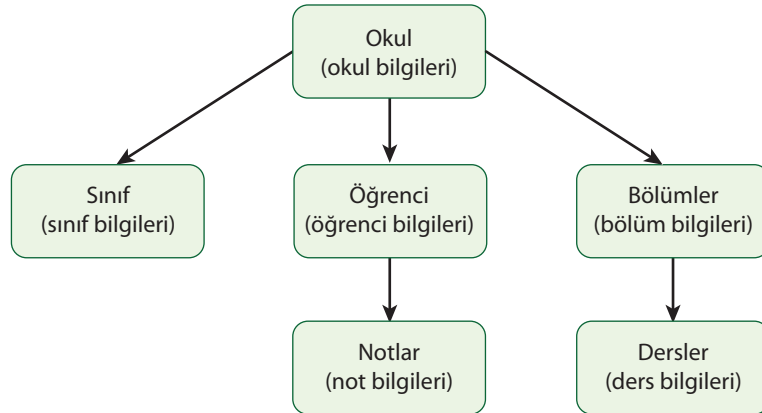
### Hiyerarşik Veritabanı

Hiyerarşik veritabanı, en eski veri modeli olan hiyerarşik veri modelini temel alıp 1960 ve 1970'li yıllarda yaygın olarak kullanılmıştır. Bu tür veritabanlarında kullanılan veri modelinde kayıtlar, ilişkileri temsil eden ve ağaç yapısına benzeyen kök ve dallar biçiminde hiyerarşik bir yapıda oluşturulur. Bu yapı, başlangıç noktası ağacın kökü, bağlantılı kılınacak noktalar dallar ve ana dallara bağlı alt dallar olarak düşünülebilen bir yapı biçimindedir. Buna göre veriler arasındaki ilişkilerde hiyerarşinin üst bölümünde olan dallar alt bölümde birden çok dal ile bağlantılı olabilirken alt bölümünde olan dallar üstte kalan dallara yalnızca tek bir noktadan bağlantılı olabilirler. Bu nedenle bu yapıda ilişkiler ebeveyn-çocuk (bir ebeveynin birden fazla çocuğu olabilirken, bir çocuğun yalnızca bir ebeveyni olması) ilişkisine benzetilir. Bu model sunucu bilgisayarlarda çalışan yazılımlar tarafından kullanılmaktadır. Bu biçimde en çok kullanılan yazılım IBM firmasının geliştirdiği IMS (Information Management System) dir. Bu yapıda gereksiz veri tekrarı söz konusu olup erişimde de sıkıntılar yaşanabilmektedir. Herhangi bir veriye erişilmek istendiğinde arama kök düğümünden başlayarak aşağı doğru devam eder. Model yapısında herhangi bir dal silinirse bu dala bağlı tüm alt dallar ve dolayısıyla ilgili tüm veriler de silinmiş olur.

**Şekil 1.4**

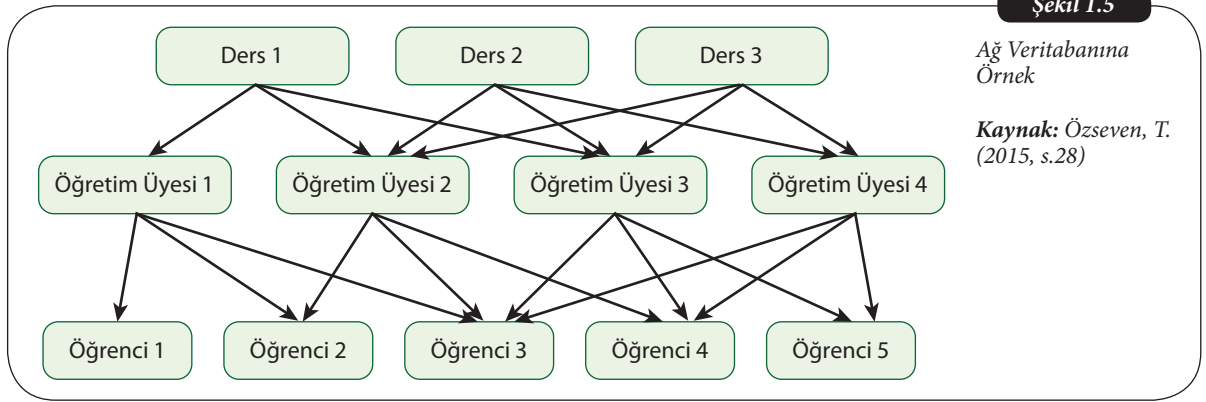
Hiyerarşik Veritabanına Örnek

**Kaynak:** Özseven, T. (2015, s.28)



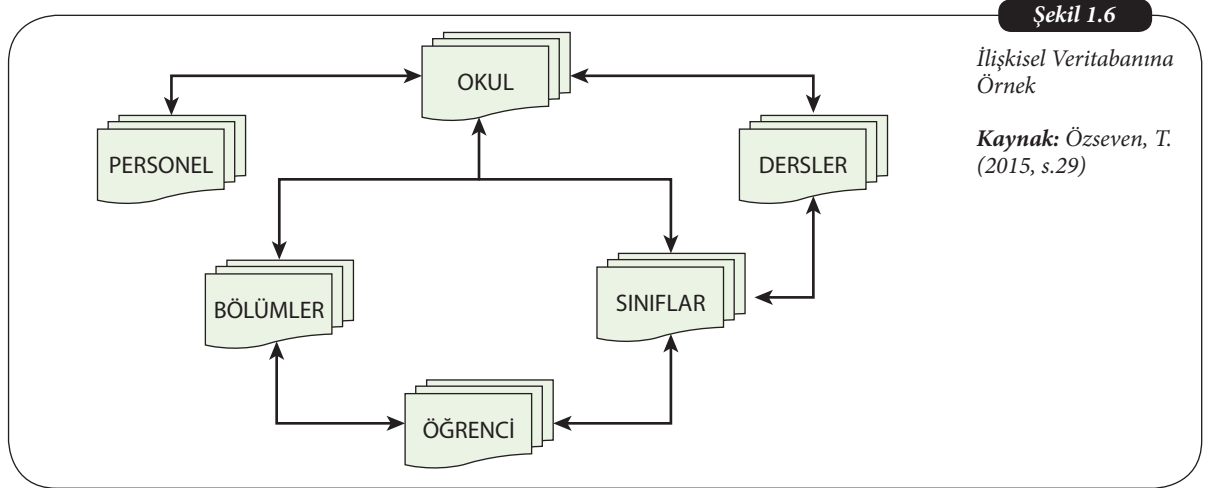
## Ağ Veritabanı

Ağ veritabanı, 1970'li yıllar ile 1980'li yılların ilk yarısında kullanılan ve ağ veri modelini temel alan veritabanı türüdür. Hiyerarşik veri modelindeki ebeveyn-çocuk ilişkisinin yetersizliği ağ veri modeliyle giderilmeye çalışılmıştır. Buna göre her bağlantı noktası düğüm olarak ifade edilirse hiyerarşik yapıdan farklı olarak ağ veri modelinde, her düğümün birden fazla ebeveyn ve birden fazla çocuk düğümü ile bağlantısı olabilir.



## İlişkisel Veritabanı

Bu tür veritabanı ilişkisel veri modelini temel almış ve ilk olarak 1970 yılında ortaya atılmıştır. 1970'li yılların sonunda kullanılmaya başlanmış ve 1985 yılından sonra kullanımı yaygınlaşmıştır. Bu yapıda ilk iki veri modelinden farklı olarak birden çok ilişki biçimi kullanılabilir. Günümüzde kullanılan veritabanı yönetim sistemlerinin hemen hemen hepsinde tercih edilen model ilişkisel veri modelidir. Bu nedenle modele ilişkin ayrıntılı bilgiler Ünite 3'te ele alınmıştır.



## Nesneye Yönelik Veritabanı

Günümüzde kullanılan ve gelecekte de kullanılacak pek çok uygulamada yalnızca harf, rakam ya da çeşitli karakterler kullanılarak yapılandırılmış verileri değil aynı zamanda multimedya (çeşitli çizim, fotoğraf, görüntü, ses ya da video gibi nesneleri) de içeren veritabanı yönetim sistemlerine ihtiyaç duyulmaktadır. Verileri satırlar ve sütunlar biçiminde düzenlemek için tasarlanmış olan veritabanı yönetim sistemleri grafik unsurları ve multimedya unsurlarını kullanmaya pek uygun değildir. Bu nedenle bu eksikliği gidermek amacıyla nesneye yönelik veri modelleri geliştirilmiştir. Söz konusu bu modelleri kullanan veritabanları da nesneye yönelik veritabanı olarak adlandırılmaktadır.

Nesneye-yönelik veri tabanları 1990'lı yıllarda kullanılmaya başlanmıştır ve giderek popüler hâle gelmektedir. Bunun nedeni, çeşitli multimedya unsurlarını ya da çeşitli kaynaklardan parça parça alınan verileri benzer biçimde birleştiren web uygulamalarında kullanılan Java uygulamalarını yönetmek için kullanılabilir olmasıdır. Nesneye yönelik veritabanları ilişkisel veri modellerinden farklı olarak daha karmaşık veri türleri üzerinde işlem yapmasına rağmen, çok sayıda işlemi yürütme açısından ilişkisel veritabanından göreceli olarak daha yavaştır. Bu nedenle günümüzde hem ilişkisel hem de nesneye yönelik veri modellerini birlikte kullanan veritabanı yönetim sistemlerinin yaygınlaştığı görülmektedir.

## VERİTABANI YÖNETİM SİSTEMİ YAZILIMLARI

Günümüzde yaygın olarak bilinen veritabanı yönetim sistemi yazılımları; MS SQL Server, Oracle, MySQL, Sybase, MS Access, PostgreSQL, IBM DB2, Informix, Advantage biçiminde sıralanabilir.

**MS SQL Server:** Microsoft firması tarafından geliştirilmiş ve ilişkisel veri modelini kullanan veritabanı yönetim sistemi yazılımıdır. Yalnızca Windows işletim sistemlerinde çalışır. Sunucu ya da istemci olarak kurulabilir.

**Oracle:** Oracle firması tarafından geliştirilmiş ilişkisel veritabanı yönetim sistemi yazılımıdır. Özellikle kurumsal amaçlı kullanılan bu yazılım büyük miktarda veriyi çok kullanıcı ortamlarda saklama ve güvenli erişim sağlama açısından, yüksek ölçekli uygulamalar için tercih edilir. Birçok işletim sistemi üzerinde kullanılabilen bu yazılımın maliyeti de yüksektir.

**MySQL:** Açık kaynak kodlu bir veritabanı yönetim sistemi yazılımıdır. Unix, OS/2 ve Windows işletim sistemlerinde ücretsiz olarak kullanılabilir. Bunun dışında ticari lisans seçeneği de mevcuttur. Özellikle web ortamında yaygın olarak kullanılan MySQL'in kullanımı kolay kullanıcı yetkilendirme sistemi güçlüdür. Python'dan Java'ya kadar birçok programlama dili ile erişilebilir.

**Sybase:** Avrupanın en büyük yazılım şirketi olan SAP'a (Systemanalyse und Programmentwicklung-Sistem Analizi ve Program geliştirme) bağlı Sybase firması tarafından geliştirilen veritabanı yönetim sistemi yazılımıdır. Orta ve büyük ölçekli uygulamalar için tercih edilmektedir. Ülkemizde daha çok bankacılık sektörü ile kamusal alanlarda kullanılmaktadır. Veritabanı üzerindeki işlemler T-SQL (Transact-SQL) sorgulama dili ile gerçekleştirilir.

**PostgreSQL:** PostgreSQL açık kaynak kodlu ücretsiz bir yazılımdır. PostgreSQL, yeni nesil veritabanı yönetim sistemi araştırma prototipi olan POSTGRES yazılımının geliştirilmiş hâlidir. POSTGRES'in zengin veri tiplerini ve güçlü veri modelini kullanırken aynı zamanda SQL'in geliştirilmiş alt kümesi olan PostQuel dilini kullanır. İyi performans veren, güvenli ve geniş özelliklere sahip bir veritabanı yönetim sistemi yazılımıdır. Hemen hemen tüm UNIX ve UNIX türevi (Linux, FreeBSD gibi ) işletim sistemlerinde çalışır. Bunun dışında Microsoft Windows NT tabanlı işletim sistemlerinde de çalıştırılabilir.

**MS Access:** Microsoft firmasının Microsoft Office yazılım paketi içinde yer alan veritabanı yönetim sistemi yazılımıdır. Küçük ölçekli uygulamalarda masaüstü veritabanı sistemi olarak kullanılır. Yalnızca Windows işletim sisteminde çalışan ve işletim sisteminin sağladığı güvenlik seçeneklerini kullanan bir yapıya sahiptir.

**DB2:** IBM firması tarafından geliştirilen veritabanı yönetim sistemi yazılımıdır. Küçük ölçekli işletmeler açısından maliyetli bir sistem olan DB2 büyük ölçekli uygulamalarda tercih edilir. Windows, Linux ve Unix işletim sistemlerinde kullanılabilir.



## Özet



### Veritabanına ilişkin temel kavramları tanımlamak

**Veri;** ham gözlemler, işlenmemiş gerçekler ya da izlenimlerdir. Bu gözlemler, gerçekler ya da izlenimler harf, rakam ya da çeşitli sembol ve işaretler yardımıyla temsil edilir. *Veritabanı* (database) herhangi bir konuda, birbiriyle ilişkili olan ve amaca uygun olarak düzenlenmiş, mantıksal ve fiziksel olarak tanımlanmış veriler bütünüdür. *Veritabanı Yönetim Sistemi* (VTYS); veritabanı tanımlamak, veritabanı oluşturmak, veritabanında işlem yapmak, veritabanının farklı kullanıcı yetkilerini belirlemek, veritabanının bakımını ve yedeklemesini yapmak için geliştirilmiş programlar bütünüdür. Veritabanı ve veritabanı yönetim sisteminin birlikte oluşturduğu bütün ise *veritabanı sistemi* olarak ifade edilir.



### Geleneksel dosya sistemleri ile veritabanı yönetim sistemlerini karşılaştırmak

Veritabanı yaklaşımından önce kullanılan ve geleneksel dosya sistemi olarak ifade edilen yaklaşımda veriler bilgisayarda ayrı ayrı dosyalar biçiminde saklanmaktaydı. Birbiriyle ilişkili olan veriler bir dosyada, başka bir açıdan birbiriyle ilişkili veriler de başka bir dosyada yer almaktaydı. Bununla birlikte iki farklı dosya içinde aynı verinin yer alması da söz konusu idi. Geleneksel dosya sistemlerinin ortaya çıkardığı sakıncalar veritabanı yönetim sistemlerinin geliştirilmesinde önemli bir etken olmuştur. Geleneksel dosya sistemlerinde büyük ölçüde veri tekrarı ve veri tutarsızlığı söz konusu olurken veritabanı yönetim sistemleri veri tekrarını ve tutarsızlığını önlemektedir. Benzer biçimde veri paylaşımına olanak vermemesi, uygulamalarda ihtiyaç duyulan değişikliklerin gerçekleştirilebilmesi için uzmanlık bilgisi gerektirmesi, istenilen veriye ulaşmada güçlükler yaşanması, verilerin güvenliği ve gizliliği konusunda sorunlar yaşanması, veriler ve uygulamalarla ilgili belirli bir standart uygulanamaması, verileri yedekleme ve kurtarma konusunda güçlükler yaşanması geleneksel dosya sistemlerinin diğer dezavantajları olarak sıralanabilir. Sıralanan bu dezavantajlar veritabanı yönetim sistemleri ile giderilebildiğinden, söz konusu bu özellikler veritabanı yönetim sistemlerinin üstünlükleri olarak karşımıza çıkmaktadır.

VTYS'ler, sağladıkları avantajlara rağmen, maliyet açısından geleneksel dosya sistemlerine göre dezavantajlıdır. VTYS için gerekli olan donanım, yazılım

ve veritabanı eğitimi için başlangıç yatırımlarının yüksek olması, veri güvenliğini sağlama, aynı anda gerçekleştirilen işlemlerin kontrolü, veri kurtarma ve bütünlük fonksiyonları için gerekli olan sabit maliyetlerin yüksek olması VTYS'nin dezavantajlı yönlerini oluşturmaktadır.

Bu nedenle, değişmesi hiç beklenmeyen basit, iyi tanımlanmış veritabanı uygulamaları söz konusu olduğunda geleneksel dosya sistemlerini kullanmak daha avantajlı olacaktır.



### Veritabanı kullanıcılarını sınıflandırmak

Veritabanı kullanıcıları veritabanı ile etkileşim biçimlerine göre sınıflandırılırlar.

*Veritabanı yöneticisi;* veritabanına erişim yetkilerini belirleme, veritabanı kullanımının düzenlenmesi ve izlenmesini sağlama, ihtiyaç duyulan yazılım ve donanım kaynaklarını edinme biçiminde sıralanan sorumluluklara sahiptir. Ayrıca güvenlik ihlalleri ve kötü sistem yanıt süresi gibi sorunların çözümünden de sorumludur.

*Veritabanı tasarımcısı;* veritabanında saklanacak olan verilerin tanımlanmasından ve bu verilerin depolanması ve gösterilmesi için gerekli olan uygun yapıların seçilmesinden sorumludur.

*Son kullanıcılar;* yaptıkları işler gereği veritabanına sorgulama ya da güncelleme yapmak veya rapor türetmek için erişen kullanıcılardır. Bu tür kullanıcılar veritabanı ile kullanıcının bağlantısını sağlayan ve uygulama programcıları tarafından geliştirilen yazılımları kullanırlar.

*Sistem analisti* son kullanıcıların, özellikle de sıradan son kullanıcıların gereksinimlerini belirleyen ve standart işlemler yoluyla bu gereksinimleri karşılayabilecek ayrıntıları belirleyen kişi ya da kişilerdir.

*Uygulama programcıları* ise sistem analisti tarafından belirlenen ayrıntıları program hâline getiren ve daha sonra test eden, hataları ayıklayan, belgeleyen ve kaydedilmiş işlemler olarak sürekliliğini sağlayan kişilerdir.



### Veritabanı yönetim sistemlerinin mimarisini açıklamak

*Veri modeli,* bir veritabanının mantıksal yapısını tanımlamada kullanılacak kavramlar, işlemler ve kurallar bütünüdür. Veritabanının mantıksal yapısı; veri tipleri, veriler arasındaki ilişkiler, veri üzerinde uygulanacak kısıtlamalar vb. dir. Veri modelleri *kavramsal veri modeli, fiziksel veri modeli ve uygulama veri mo-*



*deli* olarak sınıflandırılırlar. Veritabanının herhangi bir veri modeliyle tanımlanması ise *veritabanı şeması* olarak adlandırılır. Veritabanı yönetim sistemlerinde genel olarak üç şema mimarisi kullanılır. Üç şema mimarisinin amacı kullanıcı uygulamalarını fiziksel veritabanından ayırt etmektir. Bu mimaride şemalar izleyen üç düzeyde tanımlanır:

*İçsel (fiziksel) düzey;* veritabanının *fiziksel depolama yapısını* tanımlayan içsel şemayı içerir. İçsel şema, veriyi depolama ayrıntılarının tamamını ve veritabanına erişim yollarını tanımlayan fiziksel veri modelini kullanır.

*Kavramsal düzey;* kullanıcı topluluğu için tüm veritabanının yapısını tanımlayan kavramsal şemayı içerir. Kavramsal şema fiziksel depolama yapısının ayrıntılarını gizler ve veritabanında yer alan verilerin tipine, veriler arası ilişkilere, kullanıcı işlemlerine ve kısıtlara ilişkin tanımlara yoğunlaşır.

*Dışsal (görünüm) düzey;* bir dizi dışsal şema ya da kullanıcı görünümü içerir. Her dışsal şema bir grup kullanıcının ilgilendiği bazı veritabanı bölümlerini tanımlar. Böylece veritabanının diğer kısmı bu kullanıcı grubundan gizlenir.

Çoğu VTYS tam ve açık olarak bu üç düzeye ayrılmaz fakat bu üçlü şema mimarisini bir ölçüde destekler.

Üç şema mimarisine dayanan VTYS'lerinde her kullanıcı grubu kendi dışsal (görünüm) şemalarına başvurur. Bu nedenle VTYS, dış şemada yapılan belirli bir isteği kavramsal şema isteğine daha sonra da depolanmış veritabanı üzerinde işlem yapmak üzere içsel şema isteğine dönüştürmek zorundadır. Yapılan istek veritabanından bilgi çağırarak ise veritabanından çıkarılan veri, kullanıcının dışsal görünümüyle eşleşecek biçime dönüştürülür. Şema düzeyleri arasında bu şekilde gerçekleştirilen dönüşümler *eşleştirme* (mapping) olarak adlandırılır.

Şema düzeyleri arasındaki dönüştürme kapasitesi kısaca *veri bağımsızlığı* (mantıksal ve fiziksel) olarak ifade edilir.

VTYS'lerinde veritabanı tanımları *veri tanımlama dili* kullanılarak oluşturulur. Veritabanından veri çağırma, veri ekleme, veri silme ve veri üzerinde değişiklikler yapma faaliyetlerini gerçekleştirebilmek amacıyla kullanılan dil *veri işleme dili* olarak adlandırılırken veritabanından bilgi alma amacıyla sorgulama yapmak için kullanılan veritabanı dili ise *sorgulama dili* olarak adlandırılır.

Günümüzde kullanılan VTYS'lerinde yukarıda belirtilen dil çeşitleri genellikle birbirinden ayrı diller olarak düşünülmez aksine tüm dil çeşitlerinin görevini yerine getiren geniş kapsamlı birleştirilmiş bir dil kullanılır. Kapsamlı birleştirilmiş dile tipik örnek ilişkisel veritabanı dili *SQL (Structured Query Language-Yapısal Sorgulama Dili)*dir.



*Veritabanı türlerini ve yaygın olarak kullanılan veritabanı yönetim sistemi yazılımlarını sıralamak*

Veritabanları, kullandıkları veri modeli temel alınarak aşağıda verilen biçimde sıralanabilir:

- Hiyerarşik veritabanı
- Ağ veritabanı
- İlişkisel veritabanı
- Nesneye yönelik veritabanı

Yaygın olarak kullanılan veritabanı yönetim sistemi yazılımları ise; MS SQL Server, Oracle, MySQL, Sybase, MS Access, PostgreSQL, IBM DB2, Informix, Advantage biçiminde sıralanabilir.

## Kendimizi Sınayalım

1. Aşağıdakilerden hangisi herhangi bir konuda, birbiriyle ilişkili olan ve amaca uygun olarak düzenlenmiş, mantıksal ve fiziksel olarak tanımlanmış veriler bütününü ifade eder?
  - a. Bilgi
  - b. Bilgi deposu
  - c. Bellek
  - d. Veritabanı
  - e. Veritabanı yönetim sistemi
2. Aşağıdakilerden hangisi bilgisayar tabanlı bilgi sistemleri kapsamında kullanılan ve farklı uygulamalar tarafından paylaşılan ortak verilerin düzenlenmesi, saklanması ve kullanılması amacına yönelik yaklaşımlardan biridir?
  - a. Bilgi sistemleri
  - b. Erişim sistemleri
  - c. Uygulama programları
  - d. İşletim sistemleri
  - e. Geleneksel dosya sistemleri
3. Aşağıdakilerden hangisi veritabanında yer alan bir verinin farklı uygulamalarda kullanımı söz konusu olduğunda veri üzerinde yapılacak bir değişimin verinin kullanıldığı diğer uygulamalara da yansıtılması anlamına gelir?
  - a. Veri paylaşımı
  - b. Veri bütünlüğü
  - c. Veri gizliliği
  - d. Veri güvenliği
  - e. Veride standart yapı
4. Aşağıdakilerden hangisi veritabanı yönetim sistemlerinin avantajlarından biri **değildir**?
  - a. Gereksiz veri tekrarı ve veri tutarsızlığının önlenmesi
  - b. Veri paylaşımının sağlanması
  - c. Bakım ve güncelleme gerektirmemesi
  - d. Kullanımda üst düzey uzmanlık gerektirmemesi
  - e. Standart yapı ve kuralların uygulanabilir olması
5. Aşağıdakilerden hangisi veritabanı kullanıcıları sınıflandırmasında son kullanıcılar sınıfı içinde yer alır?
  - a. Sistem analistleri
  - b. Uygulama programcıları
  - c. Parametrik kullanıcılar
  - d. Profesyonel kullanıcılar
  - e. Veritabanı tasarımcısı
6. Aşağıdaki kavramlardan hangisi, verilerin düzenlenmesi ve depolanmasına ilişkin ayrıntıların gizlenmesi ve verinin daha iyi anlaşılmasını sağlamak için veriye ilişkin temel özelliklerin vurgulanmasını ifade eder?
  - a. Veri vurgulama
  - b. Veri soyutlama
  - c. Veri modelleme
  - d. Veri sınıflama
  - e. Veri eşleştirme
7. Aşağıdaki veri modellerinden hangisi veri modellerinin sınıflandırılmasında hem son kullanıcılar tarafından kolaylıkla anlaşılabilen kavramları hem de verinin bilgisayarda depolanması yöntemlerine ilişkin kavramları içerir?
  - a. Kavramsal veri modelleri
  - b. Yüksek düzeyli veri modelleri
  - c. Fiziksel veri modelleri
  - d. Düşük düzeyli veri modelleri
  - e. Uygulama veri modelleri
8. Aşağıdakilerden hangisi VTYS'lerinde kullanılan dillerden biri **değildir**?
  - a. Raporlama dili
  - b. Veri tanımlama dili
  - c. Veri işleme dili
  - d. Depolama tanımlama dili
  - e. Sorgulama dili
9. Veritabanındaki kayıtların ilişkileri temsil eden kök ve dallar biçiminde düzenlenen yapılarla oluşturulduğu veritabanı türü aşağıdakilerden hangisidir?
  - a. Ağ veritabanı
  - b. Hiyerarşik veritabanı
  - c. İlişkisel veritabanı
  - d. Nesneye yönelik veritabanı
  - e. Karma veritabanı
10. Aşağıdakilerden hangisi VTYS yazılımlarından biri **değildir**?
  - a. MS SQL Server
  - b. MS Access
  - c. Sybase
  - d. Unix
  - e. DB2

## Kendimizi Sınavalım Yanıt Anahtarı

1. d	Yanıtınız yanlış ise “Veritabanına İlişkin Temel Kavramlar” konusunu yeniden gözden geçiriniz.
2. e	Yanıtınız yanlış ise “Geleneksel Dosya Sistemleri ve Veritabanı Yönetim Sistemleri” konusunu yeniden gözden geçiriniz.
3. b	Yanıtınız yanlış ise “Veritabanı Yönetim Sistemlerinin Avantajları” konusunu yeniden gözden geçiriniz.
4. c	Yanıtınız yanlış ise “Veritabanı Yönetim Sistemlerinin Avantajları” konusunu yeniden gözden geçiriniz.
5. c	Yanıtınız yanlış ise “Veritabanı Kullanıcıları” konusunu yeniden gözden geçiriniz.
6. b	Yanıtınız yanlış ise “Veritabanı Yönetim Sistemlerinin Mimarisi” konusunu yeniden gözden geçiriniz.
7. e	Yanıtınız yanlış ise “Veri Modellerinin Sınıflandırılması” konusunu yeniden gözden geçiriniz.
8. a	Yanıtınız yanlış ise “Veritabanı Yönetim Sistemlerinde Kullanılan Diller” konusunu yeniden gözden geçiriniz.
9. b	Yanıtınız yanlış ise “Veritabanı Türleri” konusunu yeniden gözden geçiriniz.
10. d	Yanıtınız yanlış ise “Veritabanı Yönetim Sistemi Yazılımları” konusunu yeniden gözden geçiriniz.

## Sıra Sizde Yanıt Anahtarı

### Sıra Sizde 1

Bilindiği gibi işletmeler tüm faaliyetlerini yerine getirebilmek, farklı düzeylerde kararlar alabilmek için bilgiye ihtiyaç duyarlar. Bilgi, ham gözlemler, işlenmemiş gerçekler ya da izlenimler olarak tanımlanan **veriler** üzerinde kaydetme, sıralama, sınıflama, hesaplama, analiz vb. işlemlerin uygulanmasıyla elde edilir. İşletmelerde, üzerinde işlem uygulanarak verilecek kararlara destek sunacak biçime dönüştürülecek verilerin belirli bir düzen içinde tutulması ve istenildiği anda erişilebilmesi önemlidir. Veritabanı herhangi bir konuda, birbiriyle ilişkili olan ve amaca uygun olarak düzenlenmiş, mantıksal ve fiziksel olarak tanımlanmış veriler bütünüdür. Veritabanı işletmenin farklı bölümleri tarafından ortak olarak kullanılan verilerin, gereksiz tekrardan arındırılmış biçimde saklanması ve gerektiğinde yalnızca yetkili kişiler tarafından erişimini olanaklı kılmaktadır. Bu nedenle veritabanları verilere güvenli biçimde ve hızlıca erişim sağlaması açısından işletmelere büyük kolaylıklar sunmaktadırlar.

### Sıra Sizde 2

Geleneksel dosya sistemlerinde veriler bilgisayarda ayrı ayrı dosyalar biçiminde saklanır ve birbiriyle ilişkili olan veriler ayrı ayrı dosyalar biçiminde düzenlenirdi. Bununla birlikte iki farklı dosya içinde aynı verinin yer alması da söz konusu idi. Zaman içinde artan veri miktarı, bu verinin depolanması için gerekli olan kapasite, veriye erişim ve işleme hızında yaşanan sıkıntılar geleneksel dosya sisteminin temel sınırlılıkları olarak ortaya çıkmıştır. Bu sorunu gidermek amacıyla geliştirilen veritabanı yönetim sistemlerinin, aynı zamanda geleneksel dosya sistemlerinden ayrıldığı noktalar olan temel karakteristikleri ise aşağıdaki biçimde sıralanabilir:

- Veri tekrarı ve veri tutarsızlığını önler.
- Veri paylaşımına olanak verir.
- Uzmanlık bilgisine ihtiyaç duyulmayacak derecede kullanım kolaylığı sağlar.
- İhtiyaç duyulan veriye, tanımlanmış kullanıcı yetkileri kapsamında kolaylıkla erişilmesini sağlar.
- Veri güvenliği ve gizliliğini güçlü bir biçimde yerine getirir.
- Veriler ve uygulamalarla ilgili standart yapı ve kuralların olması kullanım kolaylığı sağlar.
- Verileri yedekleme ve kurtarma konusunda kolaylık sağlayan programlar barındırır.

### Sıra Sizde 3

Veritabanı yöneticisi; veritabanına erişim yetkilerini belirler, veritabanı kullanımını düzenler ve izler. Bunun yanı sıra ihtiyaç duyulan yazılım ve donanım kaynaklarını edinme, güvenlik ihlallerini ya da sistemin yanıt verme süresinde yaşanan sıkıntıların çözümü de veritabanı yöneticisinin sorumlulukları arasındadır.

Veritabanı tasarımcısı; veritabanında saklanacak olan verilerin tanımlanmasından ve bu verilerin depolanması ve gösterilmesi için gerekli olan uygun yapıların seçilmesinden sorumludur. Veritabanı tasarımcısı tasarım boyunca, potansiyel kullanıcı gruplarıyla karşılıklı etkileşim hâlinde olup bu grupların veriye erişimini ve veri üzerinde işlem yapabilmelerini olanaklı kılan kullanıcı görünümünü geliştirir. Her kullanıcı grubuna ilişkin görünüm, analiz edilerek diğer kullanıcı gruplarına ilişkin görünümle bütünleştirilmeleri sağlanır. Veritabanı tasarımının tüm kullanıcı gruplarının ihtiyaçlarını destekleyecek kapasitede olması gereklidir.

Yukarıda verilen ayırım keskin bir ayırım olmayıp her iki sorumluluğu aynı kişi ya da kişiler de alabilir. Veritabanı yöneticisi ile veritabanı tasarımcısının farklı kişiler olması veritabanını kullanacak olan kurumun/veritabanının büyüklüğü ve kullanıcı sayısı gibi faktörlere bağlıdır.

**Sıra Sizde 4**

Üç şema mimarisine göre veritabanlarında üç farklı düzeyde tanımlama yapılır. *İçsel şema*; veritabanının fiziksel depolama yapısını tanımlar. Bu tanımlar veriyi depolama ayrıntılarını ve veritabanına erişim yollarını içerir. *Kavramsal şema*; fiziksel depolama yapısının ayrıntılarını gizler ve veritabanında yer alan verilerin tipine, veriler arası ilişkilere, kullanıcı işlemlerine ve kısıtlara ilişkin tanımlara yoğunlaşır. *Dışsal şema*; kullanıcı görünümüyle ilgilenir. Her dışsal şema bir grup kullanıcının ilgilendiği bazı veritabanı bölümlerini tanımlar. Böylece veritabanının diğer kısmı bu kullanıcı grubundan gizlenir.

Üç şema mimarisinin amacı kullanıcı uygulamalarını fiziksel veritabanından ayırt etmektir. Üç şema mimarisine dayanan VTYS'lerinde her kullanıcı grubu yalnızca kendi dışsal (görünüm) şemalarına başvurur. Dış şemadan yapılan bu istek önce kavramsal şema isteğine daha sonra da depolanmış veritabanı üzerinde işlem yapmak üzere içsel şema isteğine dönüştürülür. Şema düzeyleri arasında bu şekilde gerçekleştirilen dönüşümler eşleştirme olarak adlandırılır. Şema düzeyleri arasındaki dönüştürme kapasiteleri mantıksal ve fiziksel veri bağımsızlığını sağlar. *Mantıksal veri bağımsızlığı*, kavramsal şemanın dışsal şemalarda ya da uygulama programlarında değişiklik yapılmaksızın değiştirilebilmesi anlamına gelir. *Fiziksel veri bağımsızlığı* ise kavramsal şemada bir değişiklik yapılmaksızın içsel şemada değişiklik yapma kapasitesidir. Buna bağlı olarak içsel şemada yapılan değişiklik dışsal şemalarda da bir değişiklik yapılmasını gerektirmez.

**Sıra Sizde 5**

Veritabanında yer alacak veriler ve veriler arasında kurulacak ilişkiler mantıksal olarak ilgili veri modeline göre yapılandırılır. Veritabanları da kullandıkları veri modeline göre sınıflandırılır.

Geçmişten günümüze kadar geliştirilmiş olan çok sayıda veri modeli, kullandıkları teknikler açısından dört temel başlıkta incelenir. Bu dört veri modelinden hangisini kullandığına bağlı olarak veritabanları da aşağıda verilen dört başlık altında sınıflandırılabilir:

- Hiyerarşik veritabanı
- Ağ veritabanı
- İlişkisel veritabanı
- Nesneye yönelik veritabanı

## Yararlanılan ve Başvurulabilecek Kaynaklar

- Aydın, S. (2013). **İşletme Bilgi Sistemleri**, (Editör: H. Durucasu), Eskişehir: Anadolu Üniversitesi Yayın No:2690, AÖF Yayın No:1656.
- Barutçugil, İ. (2002). **Bilgi Yönetimi**, İstanbul: Kariyer Yayıncılık İletişim.
- Burma, Z.A. (2009). **Veritabanı Yönetim Sistemleri ve SQL/PL-SQL/T-SQL**, Ankara: Seçkin Yayıncılık.
- Elmasri, R. ve Navathe, S.B. (2011). **Fundamentals of Database Systems**, USA: Pearson Education, Inc.
- Laudon, K.C. ve Laudon J.P. (2011). **Yönetim Bilişim Sistemleri Dijital İşletmeyi Yönetme**, Çeviri Editörü: U.Yozgat, Ankara: Nobel Akademik Yayıncılık.
- O'brien, J.A., ve Marakas M. (2010). **Introduction to Information Systems in Business Management**, Fifteenth Edition, New York:McGraw-Hill Irwin.
- Öğüt, A. (2003). **Bilgi Çağında Yönetim**, Ankara: Nobel Yayın Dağıtım.
- Özseven, T. (2015). **Veritabanı Yönetim Sistemleri I**, Bursa: Ekin Basım Yayın Dağıtım.
- Vural, Y. Ve Sağıroğlu, Ş. (2010). *Veritabanları Yönetim Sistemleri Güvenliği: Tehditler ve Korunma Yöntemleri*, Politeknik Dergisi, Cilt:13, Sayı:2 s.71-81.
- Yarımağan, Ü. (2000). **Veritabanı Sistemleri**, Ankara: Akademi Yayın Hizmetleri San.Tic. Ltd.Şti.
- <http://quickbase.intuit.com/articles/timeline-of-database-history> (erişim tarihi: 26.11.2015)
- <http://www.microsoft.com/tr-tr/server-cloud/products/sql-server-editions/overview.aspx> (erişim tarihi: 17.11.2015)
- <http://www.mysql.com.tr/tr/KonuDetay.php?AKey=115&BKey=34&CKey=50> (erişim tarihi: 17.11.2015)
- [http://www.postgresql.org/message-id/attachment/17832/FAQ\\_turkish.html#1.1](http://www.postgresql.org/message-id/attachment/17832/FAQ_turkish.html#1.1) (erişim tarihi: 17.11.2015)
- Gündüz, D. <http://ab.org.tr/ab04/tammetin/178.pdf>