

# VERİTABANI YÖNETİM SİSTEMLERİNE GİRİŞ

Dr. Öğr. Üyesi Ender Şahinaslan

# BÖLÜM -1-

## TEMEL KAVRAMLAR



# GENEL BAKIŞ...

---

- Giriş,
- Veritabanı temel kavramları,
- Veri erişim yöntemleri,
- Veritabanı yönetim sistemleri,
- Neden veritabanı yönetim sistemleri tercih edilmeli?
- Veritabanı türleri,
- Veritabanı kullanıcıları,
- SQL nedir?
- Veritabanı Oluşturma Aşamaları

# 1. GİRİŞ

- Günümüzde işletmelerin çoğu varlıklarını sürdürebilmek için bilgiye ihtiyaç duyarlar.
- İşletme içerisinde dolaşan bilginin kalitesi oldukça önemlidir.
- Bilginin zamanında elde edilebilmesi, ilgili kişi ve birimlere ihtiyaç duyulduğu anda iletilebilmesi bilgi sistemleri aracılığıyla gerçekleştirilir.



# 1. GİRİŞ (DEVAM...)

- Bilgi sistemleri;** ilgili verilerin toplanması, bu veriler üzerinde birtakım işlemler uygulanarak verilerin **yararlı bilgilere dönüştürülmesi**, ilgili kişi ve birimlere **iletilmesini**, belirli bir düzen içinde gerçekleştiren sistemlerdir.



# 1. GİRİŞ (DEVAM...)

---

- Veritabanı sistemleri kısaca veri kaynaklarının yönetiminden sorumlu sistemlerdir.
- Veritabanı sistemlerine ilişkin ayrıntılı bilgilerin yer aldığı bu bölümde veritabanı sistemlerine ilişkin temel kavramlar ve genel bilgiler ele alınmıştır.

# 1.1. VERİTABANINA İLİŞKİN TEMEL KAVRAMLAR

---

- **Veritabanı** çok genel anlamda, bir kurumun ihtiyaç duyduğu ve kullandığı veriler bütünüdür.
- Bu noktada öncelikle veri ve bilgi kavramları 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.
  - **Bilgi** ise yalın tanımıyla verinin işlenmiş ve karar verme sürecine destek olacak duruma dönüştürülmüş biçimidir.

# 1.1. VERİTABANINA İLİŞKİN TEMEL KAVRAMLAR

- **Veritabanı**, herhangi bir konuda birbirleriyle ilişkili olan ve amaca uygun olarak düzenlenmiş, mantıksal ve fiziksel olarak tanımlanmış veriler bütünüdür.
- Veritabanı herhangi bir kurumda birden fazla uygulamada ortak olarak kullanılabilen verilerden oluşur.
- Veritabanı içerisinde sürekli niteliği olan veriler yer alır.





# 1.1. VERİTABANINA İLİŞKİN TEMEL KAVRAMLAR

- ***Veritabanı Yönetim Sistemi;*** veritabanı'nı tanımlamak, oluşturmak, işlem yapmak, farklı kullanıcı yetkilerini belirlemek, bakım ve yedekleme işlemlerini yapmak için geliştirilmiş programlar bütünüdür.
  - ***Veritabanı Tanımlamak:*** Veritabanında yer alacak verinin adı, tipi, uzunluğu gibi veri yapıları ve özelliklerinin belirlenmesidir.
  - ***Veritabanı Oluşturmak:*** Veritabanını oluşturma ve veriyi depolama sürecidir.
  - ***Veritabanında İşlem Yapmak:*** Veritabanını izleme, veriler üzerinde sorgulama yapma, veritabanında gerekli olan değişiklikleri yaparak veritabanını güncelleme ve verilerden rapor elde etme işlevlerini içerir.

## 1.2. VERİ ERİŞİM YÖNTEMLERİ

---

- Geçmişten beri veriye erişim amacıyla farklı yaklaşımlar kullanılmıştır.
- Bu yaklaşımlardan üçü; sıralı erişim, doğrudan erişim ve hesaba dayalı erişim biçimindedir.



## 1.2.1. GELENEKSEL DOSYA YAPILARI

---

Hem donanım hem de yazılım alanında **pek çok değişim** ortaya çıkmıştır. Bu değişim, kuruluşların ve kişilerin **gereksinimlerine daha iyi cevap verebilmek** amacıyla ortaya çıkmaktadır.

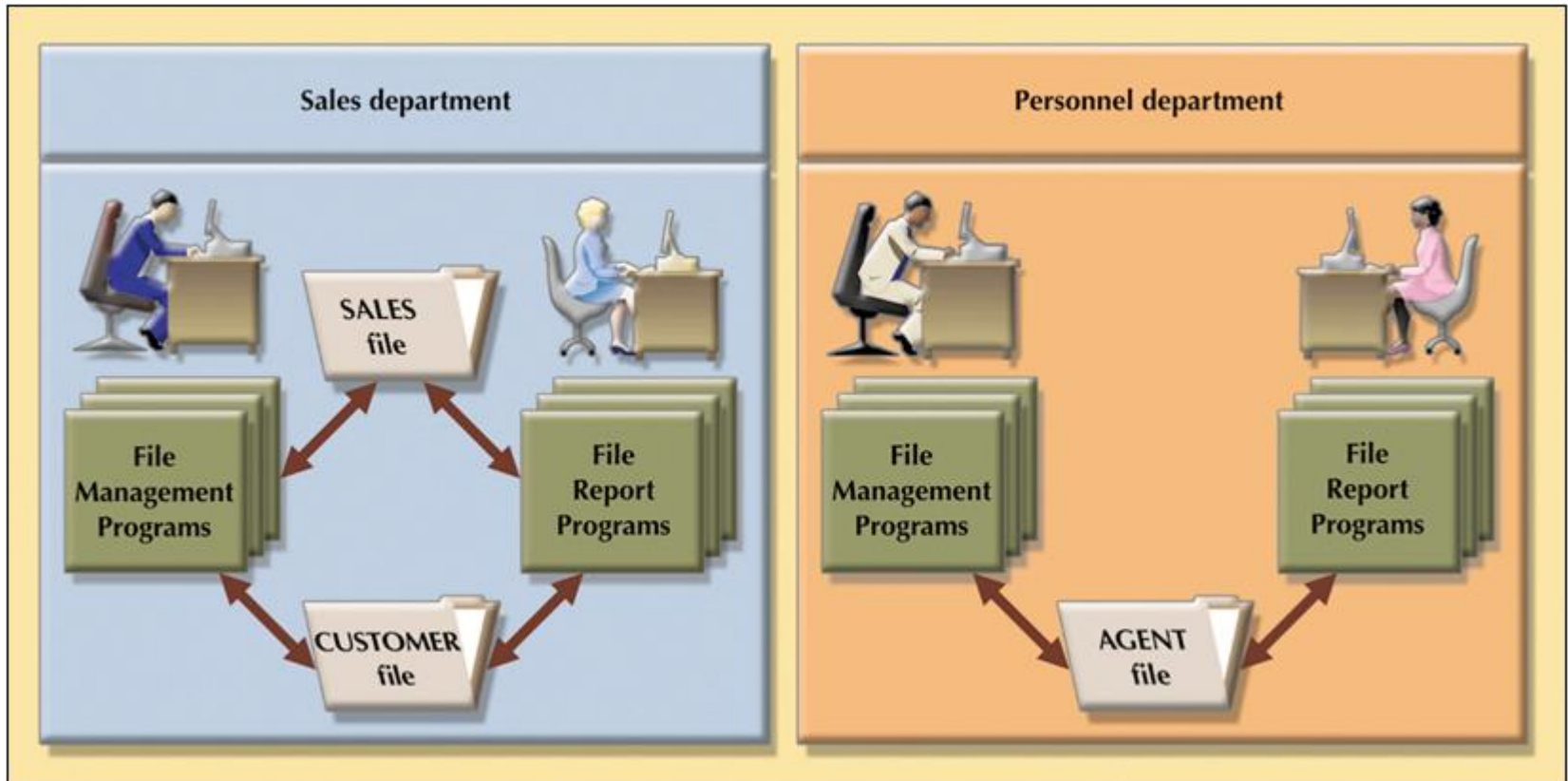
Pek çok kuruluştaki **eskiye oranla çok daha fazla veri üretildiğinden**, geleneksel **veri saklama** ve **işleme** tekniklerinin yetersiz kaldığı gözlemlenmiştir.

Veri saklama birimlerinde depolanan veri topluluklarına **“dosya”** (*file*) denir.



## 1.2.1. GELENEKSEL DOSYA YAPILARI (DEVAM)

Aşağıda basit bir klasik dosya sistemi görülmektedir.



## 1.2.1. GELENEKSEL DOSYA YAPILARI (DEVAM)

---

Geleneksel dosya sistemlerinin sakıncaları:

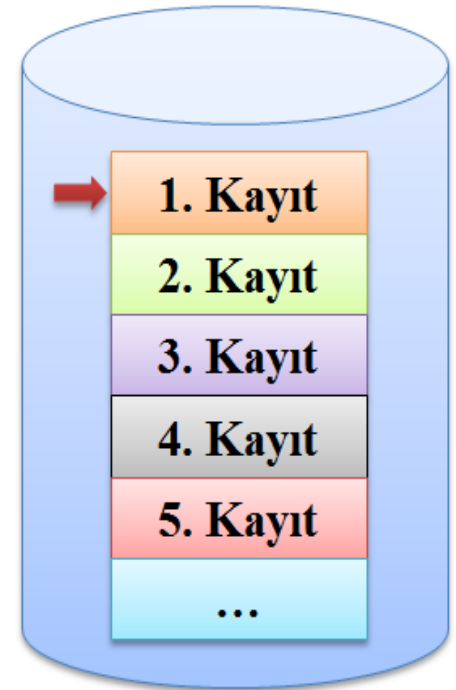
- Veri tekrarı ve tutarsızlığına yol açar.
- Veri paylaşımına olanak vermez.
- Uygulamalarda ihtiyaç duyulan değişikliklerin gerçekleştirilebilmesi için uzmanlık bilgisi gerektirir.
- İstenilen veriye ulaşmada güçlüklerle karşılaşılır.
- Verilerin güvenliği ve gizliliği konusunda sorun yaşanır.
- Verileri yedekleme ve kurtarma konusunda güçlükler yaşanır.

## 1.2.2. SIRALI DOSYALAR

Sıralı dosyalar, bir başka deyişle ardışık dosyalar, içerdiği kayıtlara birinci kayıttan başlamak üzere sırayla erişim yapmak üzere tasarlanmış dosyalardır.

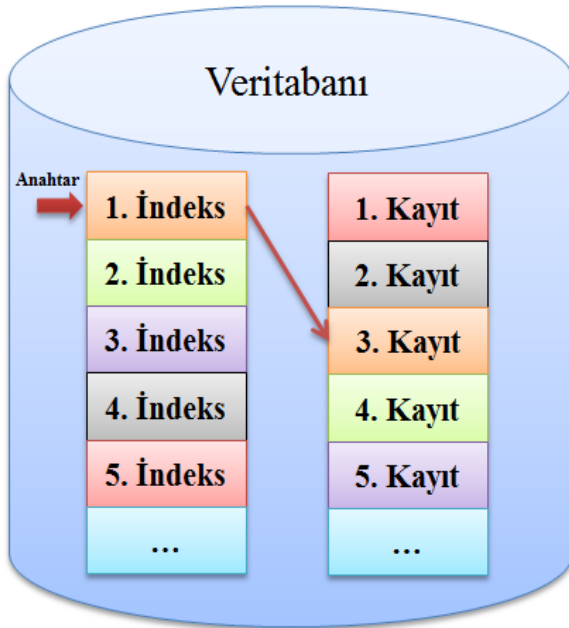
Sıralı dosyaların her bir kayıtına ardışık olarak erişilmesi bazı durumlarda yararlı olmasına rağmen, bazı uygulamalarda sorunlar yaratır.

Sıralı erişimde, istenilen veriye ulaşıncaya kadar ilgili dosyadaki tüm verilerin sırayla okunması gerekir.



## 1.2.3. İNDEKSLİ DOSYALAR

Doğrudan erişimli dosyaların en tanınmış, indeksli dosyalar olarak bilinir. İndeksli dosyalar, veri dosyasından ayrı olarak bir indeks dosyasının oluşturulması ile birlikte hazırlanır.



Bir dosya için oluşturulan indeks; söz konusu dosyanın anahtarları ile bu anahtarların disk üzerinde bulunduğu adresi içerir.

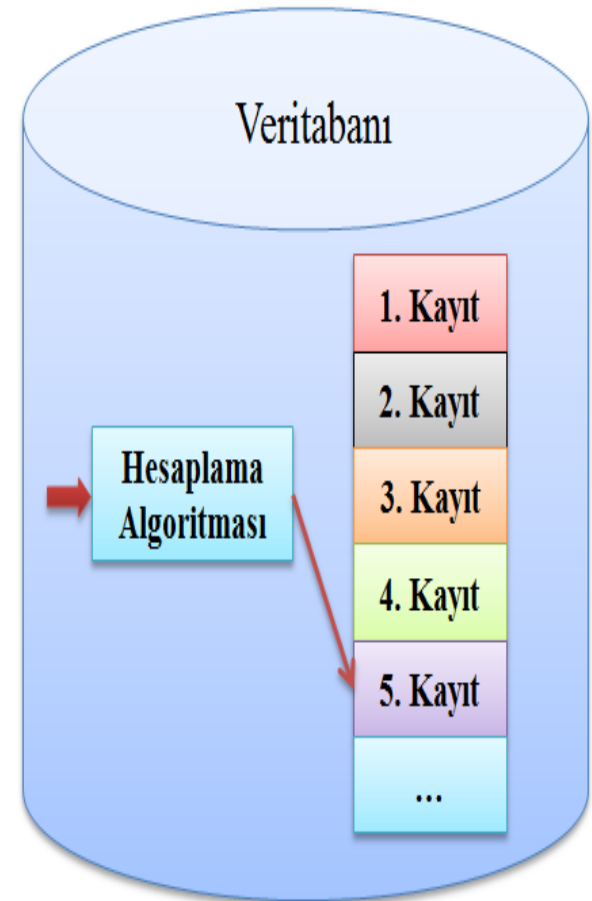
Anahtar alan, erişimde kullanılmak üzere seçilen alan olarak değerlendirilir.

**Hatırlatma:** İndeksli dosyalarda önce indeks dosyasından bir kayıt okunur. Bu kayıt veri dosyasının ilgili satırına ait adresi içerir. Bu adres ile ilgili kayıt satırına ulaşılır.

## 1.2.4. HESABA DAYALI DOSYALAR

Bu tür dosyalar, indeksli dosyalar gibi ayrı bir indeksin tutulmasını gerektirmez.

Dosyanın herhangi bir kaydına doğrudan doğruya erişebilmek için bir hesaplama algoritması kullanır.





## 1.3. VERİTABANI YÖNETİM SİSTEMLERİ

Geleneksel dosya sisteminin yetersiz kaldığı durumlardaki sorunu çözmek üzere, **veriyi saklama ve veriye erişim konusunda** yeni yazılım teknolojilerine yönelme başlamış ve Veritabanı Yönetim Sistemleri (VTYS) yaklaşımı ortaya çıkmıştır.



## 1.3. VERİTABANI YÖNETİM SİSTEMLERİ (DEVAM...)

---

Veritabanı yönetim sistemleri (VTYS) bilgisayar sistemlerinin önemli bir bileşeni olarak değerlendirilir.

VTYS, birbirleriyle ilişkili veri ve programlar topluluğundan oluşmaktadır.

Veri topluluğu bir veritabanı olarak değerlendirilir.

Veritabanı bir kuruluşa ilişkin bilgilerin yer aldığı ortamdır.

VTYS, veri kümelerinin düzenli biçimde tutulduğu ve bu verilerin çeşitli yazılımlar aracılığıyla yönetildiği ortamlardır.

## 1.3. VERİTABANI YÖNETİM SİSTEMLERİ (DEVAM...)

---

Piyasa da yaygın olarak kullanılan başlıca veri tabanı yönetim sistemi yazılımları:

- Oracle,
- Microsoft SQL Server,
- PostgreSQL
- Sysbase,
- DB2,
- MySQL
- Paradox,
- Firebird,
- Filemaker,
- BerkeleyDB,
- Informix
- Microsoft Access.

## 1.3. VYTS'NİN SAĞLADIĞI ÜSTÜNLÜKLER

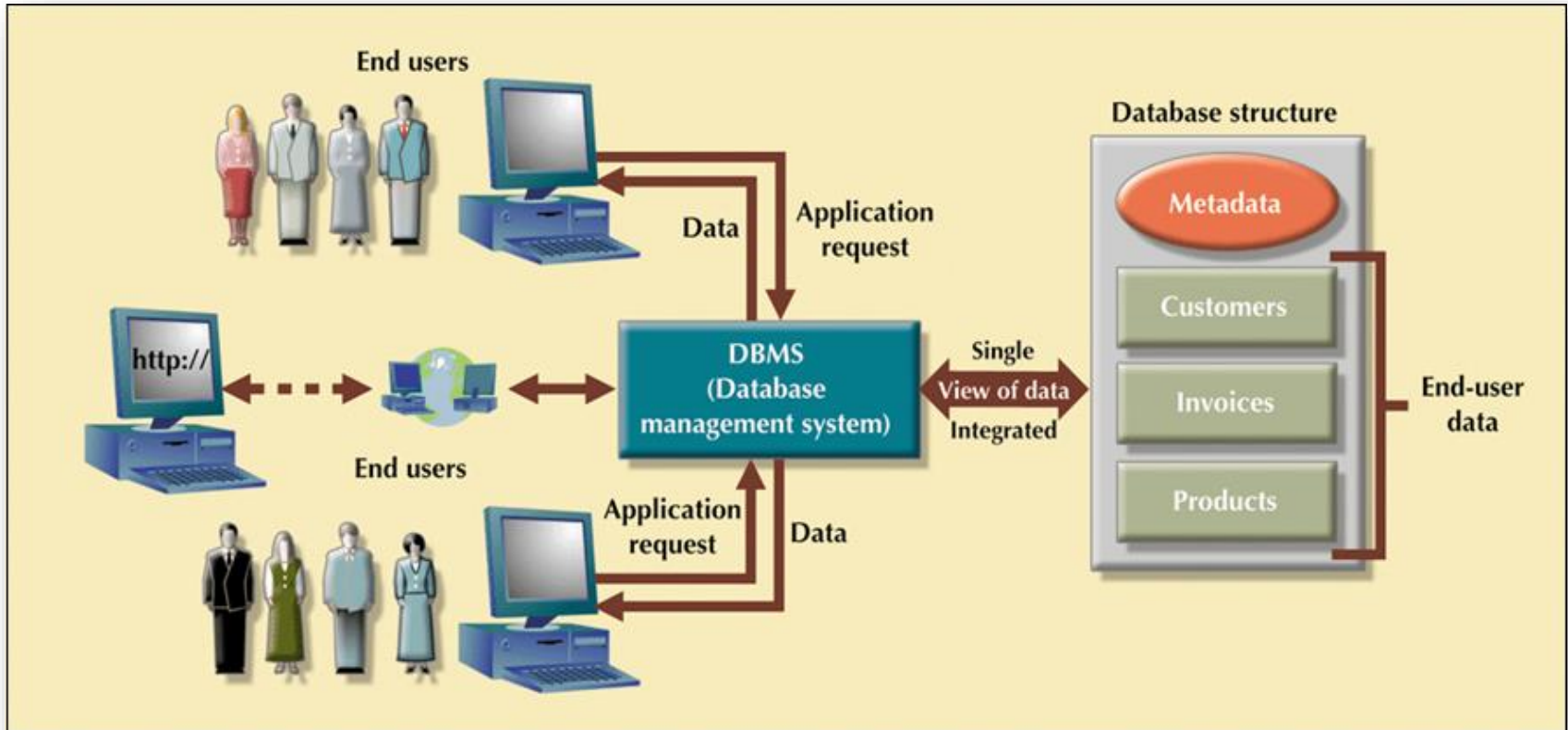
---

Veritabanı kullanımı, geleneksel dosya kullanımına göre birçok yönden üstünlük sağlamaktadır. Bunlar:

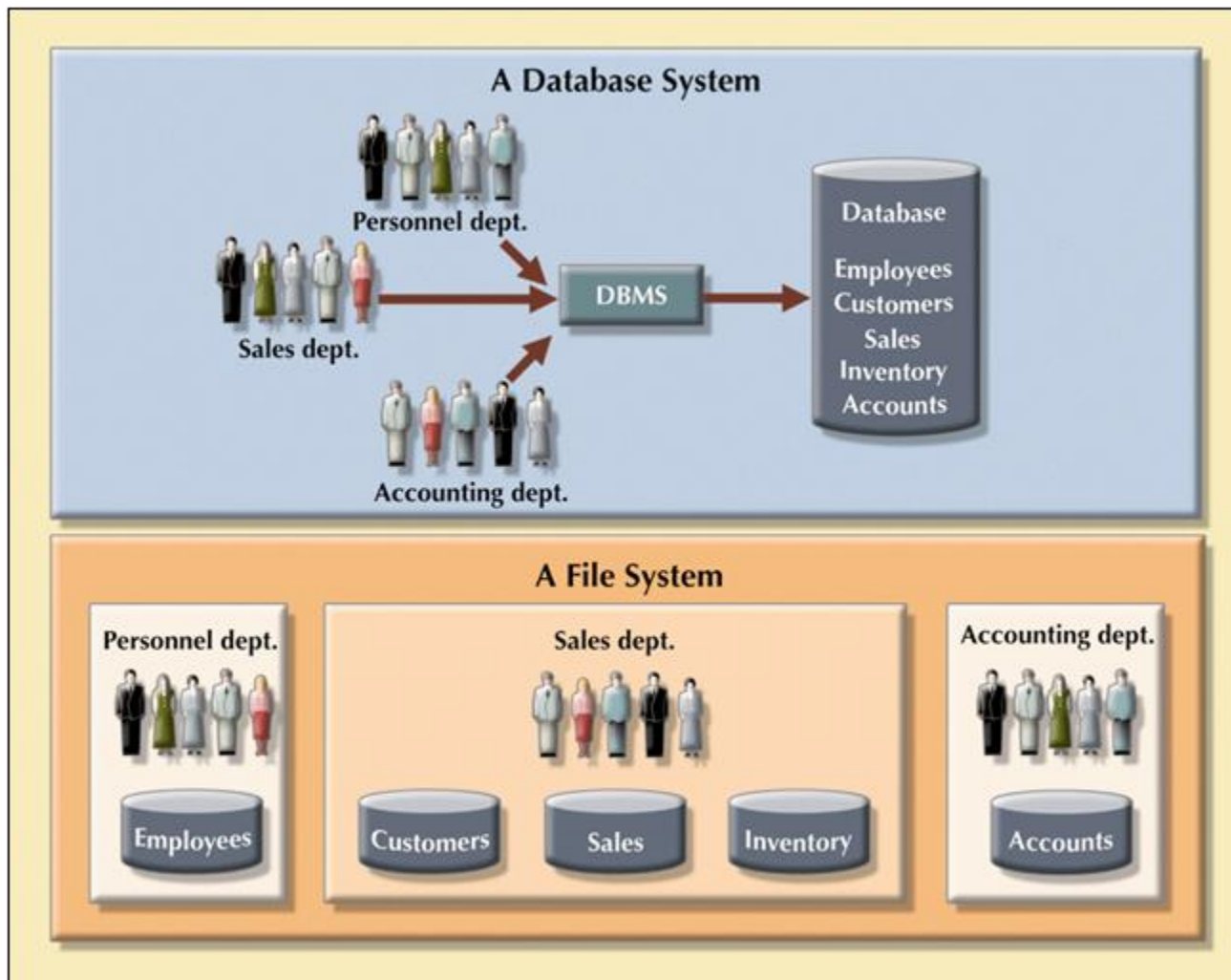
- Verinin tekrarlamasını önler,
- Verinin tutarlı olmasını sağlar,
- Aynı andaki erişimlerde tutarsızlıkların ortaya çıkmasını önler.
- Verinin güvenliğini sağlar.

## 1.3. VERİTABANI YÖNETİM SİSTEMLERİ (DEVAM...)

VTYS, son kullanıcı ile veritabanı arasındaki etkileşimi yönetir.



## 1.3. VERİTABANI YÖNETİM SİSTEMLERİ (DEVAM...)



## 1.3.1. VTYS'NİN SAĞLADIĞI YARARLAR

---

- VTYS kullanımının sağladığı faydaları şöyle sıralayabiliriz:
  - + Veri bağımsızlığı,
  - + Etkin veri erişimi,
  - + Yetkisiz veri girişlerini engelleme,
  - + Veri yönetimi,
  - + Veriler arasında karmaşık ilişkiler tanımlama olanağı,
  - + Eşzamanlı kullanım,
  - + Hata durumunda sistem geri yükleme,
  - + Güncel veriye anında erişim imkanı sağlama,
  - + Uygulama geliştirme zamanının kısaltılması.

## 1.3.2. VTYS KULLANIMININ DEZAVANTAJLARI

---

- VTYS kullanımının klasik dosya sistemine göre dezavantajlarını şöyle sıralayabiliriz:
  - ✦ Maliyetin yüksek olması (lisans, kurulum, programlama, bakım vs.),
  - ✦ Bir arıza durumunda veri dönüşümü daha zordur,
  - ✦ Bazı durumlarda uygulamanın yavaş çalışmasına neden olabilir,
  - ✦ Karmaşık yapısı nedeniyle kırılganlığı daha yüksektir, yani yolunda gitmeyen problemler ciddi sıkıntılara yol açabilir.



## 1.4. VERİTABANI TÜRLERİ

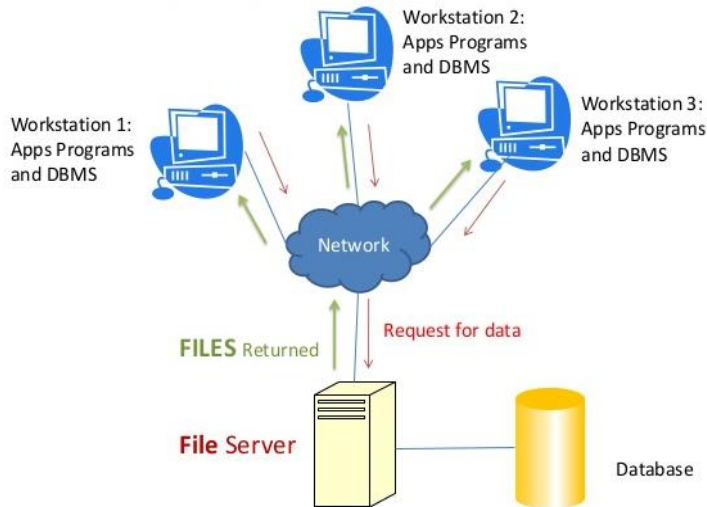
---

Veritabanları;

- + kullanıcı sayısına,
- + fiziksel konumuna,
- + veri modeline göre üç şekilde sınıflandırılabilir.

## 1.4.1. KULLANICI SAYISINA GÖRE

- Tek-kullanıcılı (single-user) veritabanları, bir seferde yalnızca bir kullanıcıyı destekler.
  - ✚ Masaüstü veritabanı: tek-kullanıcılı, PC üzerinde çalışır.

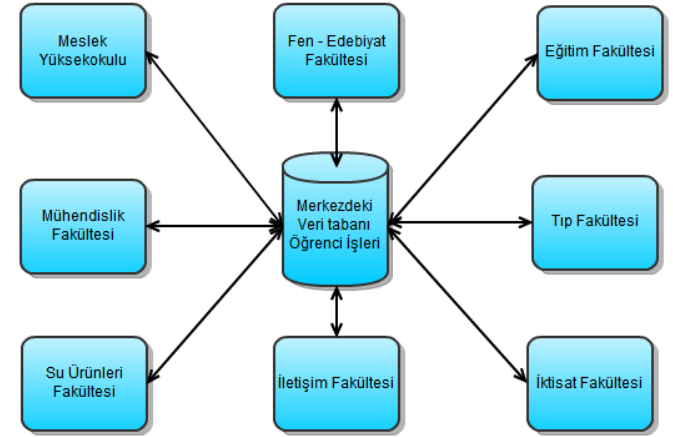


Çok-kullanıcılı (multi-user)  
veritabanları, aynı anda birden  
çok kullanıcıyı destekler.

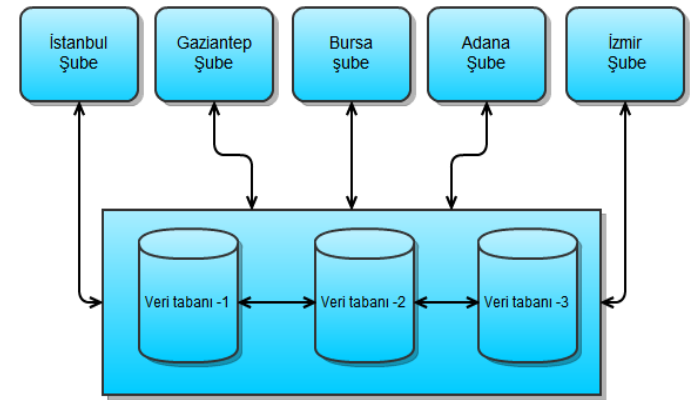
Kurumsal veritabanları...

## 1.4.2. FİZİKSEL KONUMUNA GÖRE

**Merkezi Veritabanı:** Tek bir veritabanı sistemi bulunur. Fiziksel olarak ta tek bir merkezde bulunur.



- **Dağıtık Veritabanı:** VTYS'nin çeşitli parçaları bir çok bölgeye dağıtılmıştır. Dağıtılmış veritabanı, kullanıcıya tek bir veritabanı gibi gözükür, fakat ayrı yerlerdeki veritabanlarından oluşur.



### *1.4.3. VERİ MODELİNE GÖRE*

---

Bir veritabanı yapısının temelini **veri modeli** kavramı oluşturmaktadır.

Veriyi mantıksal düzeyde düzenlemek için; kullanılan kavramlar, yapılar ve işlemler topluluğuna **veri modeli** denir.

Birçok veri modeli geliştirilmiştir. Başlıca veri modelleri;

- + **Sıra düzensel** veri modeli,
- + **Ağ** (network) veri modeli,
- + **İlişkisel** veri modeli,
- + **Nesneye yönelik** veri modeli.

### 1.4.3.1. SIRADÜZENSEL (HİYERARŞİK) VERİ MODELİ

---

Hiyerarşik veri modeli 1960 ve 1970 yılları arasında popüler olan bir modeldir. Bu modelde çoklu ilişkileri temsil edebilmek için, varlık tiplerinin her ilişki için ayrı ayrı tanımlanması gerekir. Bu da gereksiz veri tekrarına sebep olur.

Bu modelde, veriler ağaç yapısına benzer bir biçimde modellenir. En üste kök ve kökün dalları bulunur. Ayrıca her dalın alt dalı sayesinde dallanma ve çeşitlilik artar.

Bu modelde her bir alt dalın sadece bir tane noktadan bağlanma şartı bu modelin en büyük kısıtlamalarından biridir.

### 1.4.3.1. SIRADÜZENSEL (HİYERARŞİK) VERİ MODELİ (DEVAM...)

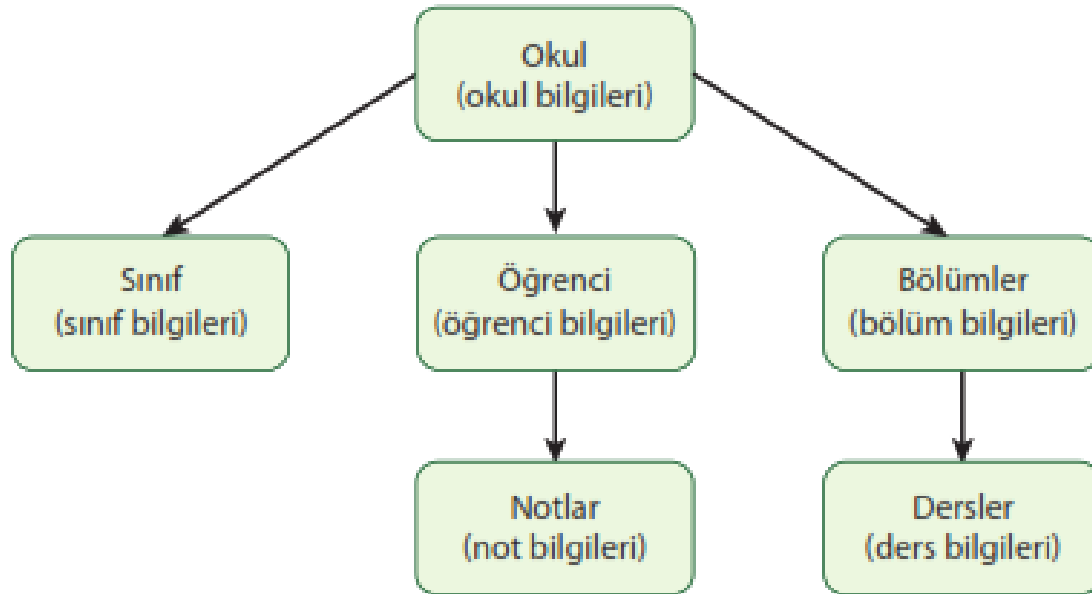
---

Model dahilindeki herhangi bir düğüm, altındaki  $n$  sayıda düğüme bağlanırken, kendisinin üstünde ancak bir düğüme bağlanabilir.

Hiyerarşik yapının en tepesindeki düğüm noktasına **kök** denir ve bu düğümün sadece bağımlı düğümleri bulunur. Bu veri yapısını gösteren grafiğe de **hiyerarşik tanım ağacı** denir.

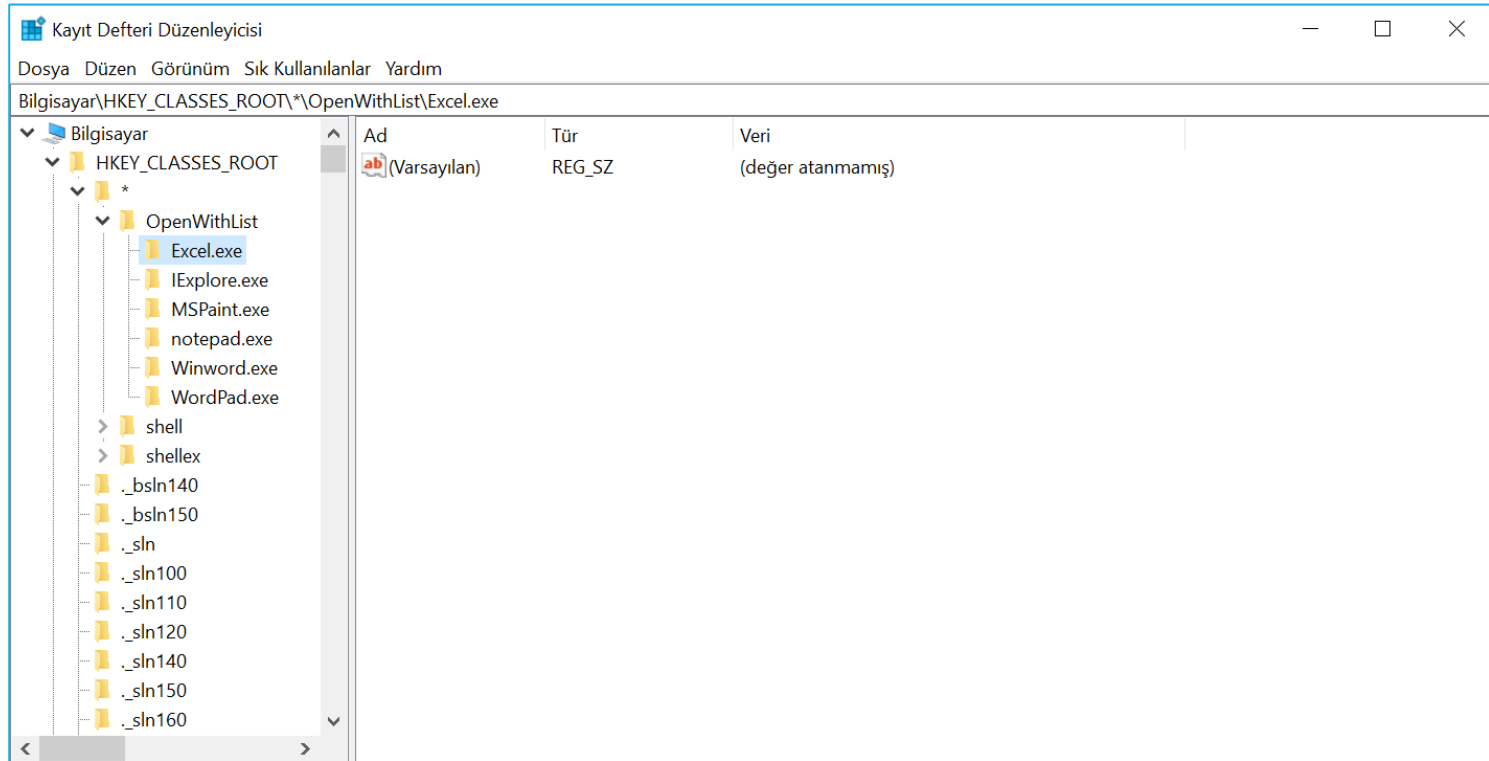
### 1.4.3.1. SIRADÜZENSEL (HİYERARŞİK) VERİ MODELİ (DEVAM...)

Bir hiyerarşik tanım ağacının örneği aşağıda görülmektedir.



## 1.4.3.1. SIRADÜZENSEL (HİYERARŞİK) VERİ MODELİ (DEVAM...)

Hiyerarşik veri modeline en güzel örnek bilgisayarın registry kayıtlarıdır (regedit).





### 1.4.3.2. AĞ (NETWORK) VERİ MODELİ

---

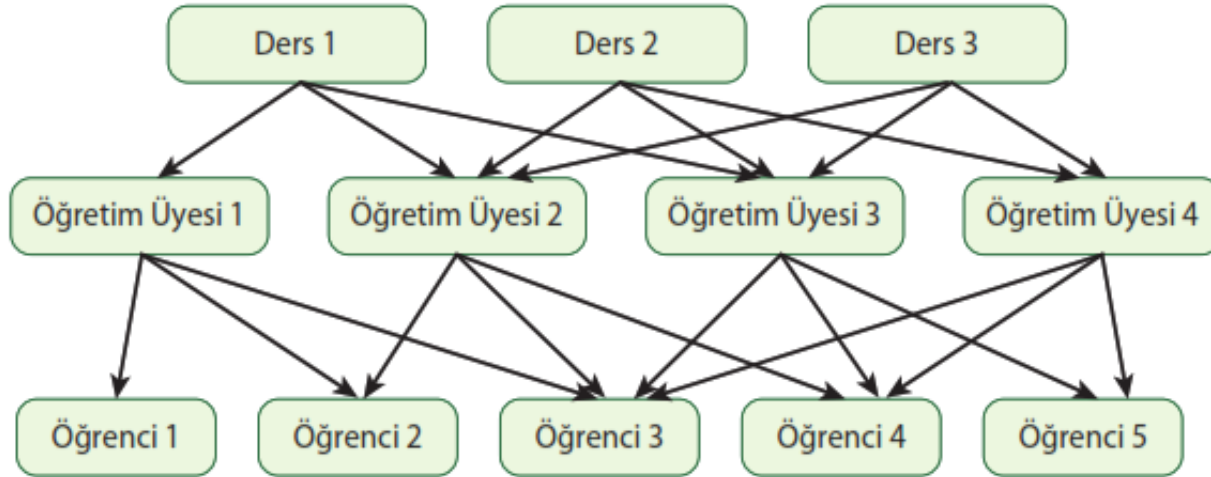
Ağ veri modeli 1970'li yılların başında geliştirilmiştir. Bir verinin doğası gereği birden çok veri ile ilişkisinin olmasından dolayı hızlıca kabul görmüştür.

Bu modelde verilerin birbirine ağ şeklinde bağlandığı varsayılır.

Bir veri ağında iki kullanıcının haberleşebilmesi için bir çeşit adresleme şekli gereklidir. Genelde, iki veya üç adres gerekir. Bazı sistemlerde bir fiziksel adres, bir veri bağlantı adresi ve bir ağ adresi kullanılır. Daha yaygın bir yaklaşım, yalnızca fiziksel ve ağ adreslerini kullanmaktır. Bu yaklaşımda, fiziksel ve veri bağlantı adresi aynıdır. Veri iletişimi bu adresler aracılığı ile sağlanmaktadır.

### 1.4.3.2. AĞ (NETWORK) VERİ MODELİ

Bir ağ veri modeli örneği aşağıda görülmektedir.



### 1.4.3.3. İLİŞKİSEL VERİ MODELİ

---

Şu anda kullanılan veri tabanlarının çoğu ilişkisel veri modelini desteklemektedir.

Bu modelde birbiri ile alakalı olan veriler, tablolar içinde saklanır. Ayrıca tablolar arasında değişik türlerde ilişki kurulmaktadır.

İlişkiler kurulurken birincil anahtar (primary key) ve yabancı anahtarlar (foreign key) kullanılır.

Anahtar alanlar sayesinde indeksleme yapma olanağı sunan ilişkisel veri tabanlarında erişim ve işlemler daha hızlı yapılabilmektedir.

İlişkiler ve onların temsilleri olan tablolardan oluşan veri modelleri ilk olarak 1970 yılında Codd tarafından ortaya atılmıştır.

### 1.4.3.3. İLİŞKİSEL VERİ MODELİ (DEVAM...)

---

İlişkisel veri modelleri formüle edilirken, veri yönetimi ihtiyaçlarını karşılayabilmek için ilişkinin matematiksel teorisi, mantıksal olarak genişletilmiştir.

İlişkisel veri modellerinde kullanılan tek yapılandırma aracı ilişkidir. İlişkinin tanımı, veri tabanı ilişkilerinin zamana bağlı olması dışında, matematiksel tanımı ile aynıdır. Yani, bir veri tabanı ilişkisinde satırlar; eklenebilir, değiştirilebilir ya da düzeltilebilir.

### 1.4.3.3. İLİŞKİSEL VERİ MODELİ (DEVAM...)

---

Aşağıda gördüğünüz satırlar, basit bir hastane veri tabanının ilişkisel şemasını göstermektedir.

**HASTANE (Hastane\_Kodu, Hastane\_Adi, Adres, Tel\_No, Yatak\_Sayısı)**

**POLIKLINIK (Poliklinik\_No, Poliklinik\_Adi, Hastane\_Kodu)**

**DOKTOR (Hastane\_Kodu, Diploma\_No, Adı, Soyadı, Uzmanlık\_Alanı)**

#### 1.4.3.4. NESNEYE YÖNELİK VERİ MODELİ

---

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.

#### 1.4.3.4. NESNEYE YÖNELİK VERİ MODELİ (DEVAM...)

---

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.

#### 1.4.3.4. NESNEYE YÖNELİK VERİ MODELİ (DEVAM...)

---

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.

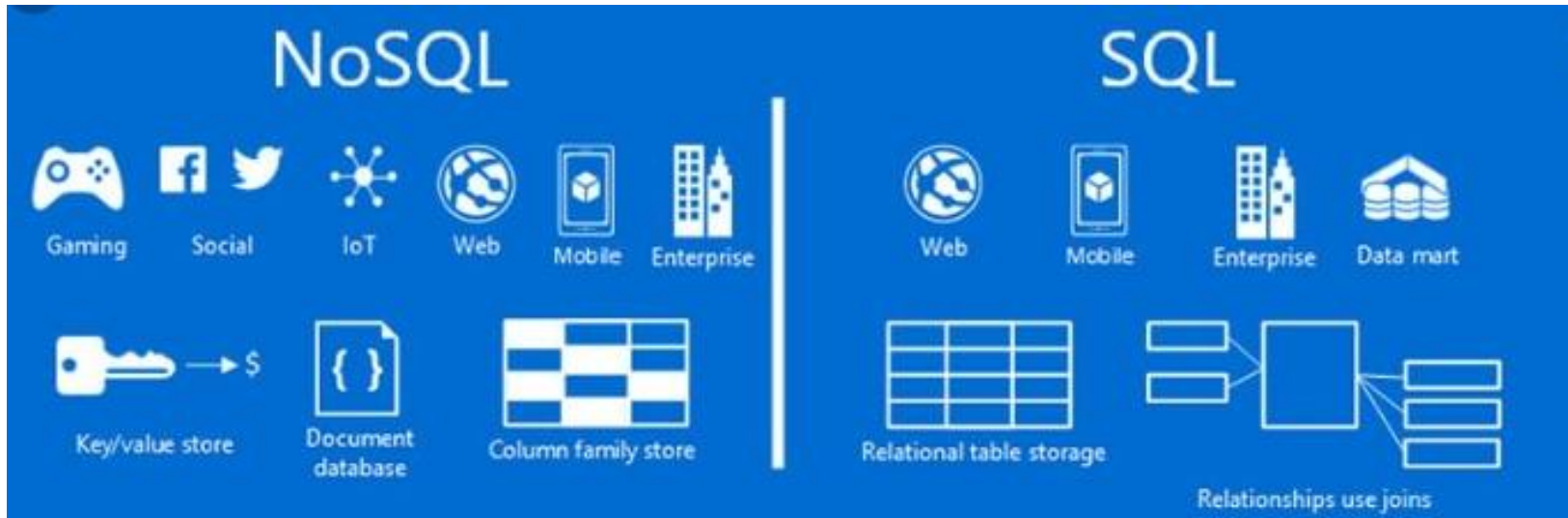


## 1.4.4. DİĞER VERİ TABANI TÜRLERİ

Veri ambarları

NoSQL veritabanları

- Grafik veritabanları
- OLTP veritabanları



## 1.4.4.1. NOSQL VERİ TABANLARI

---

### Anahtar-Değer (Key-Value)

– MemcacheDB , Redis , Amazon DynamoDB, Riak, Voldemort

- Doküman-Tabanlı (Document-Oriented)

– MongoDB, CouchDB , RavenDB, MarkLogic

- Sütun-Tabanlı (Column-Oriented)

– Hbase, Cassandra, Accumulo

- Çizelge (Graph)

– Neo4J, OrientDB, Allegro, Virtuoso, Sones, Jena, Sesame



## 1.5. 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
- Sistem Analistleri ve Uygulama Programcıları

## 1.5.1. VERİTABANI SORUMLULARI

Veritabanı sorumluları; veritabanının tasarlanması, oluşturulması ve işletim faaliyetlerinden birinci derecede sorumlu olan ve veritabanı üzerinde en fazla yetkiye sahip olan kullanıcılardır.

Veritabanı sorumluları, veritabanı yöneticisi ve veritabanı tasarımcısı olarak iki başlık altında incelenebilir.



## 1.5.1. VERİTABANI SORUMLULARI (DEVAM...)

---

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.

Veritabanı tasarımcısı (database designer) veritabanında saklanacak olan verilerin tanımlanmasından ve bu verilerin depolanması ve gösterilmesi için gerekli olan uygun yapıların seçilmesinden sorumludur.

## 1.5.1. VERİTABANI SORUMLULARI (DEVAM...)

---

Veritabanı sorumlularının yerine getirdikleri temel görevler:

- Veritabanı tasarımını yapma,
- Bütünlük kısıtlamalarını belirleyip tanımlama,
- Veritabanı kullanım yetkilerini tanımlama,
- Veritabanı güvenliğini sağlama,
- Veritabanının işletimini izleme ve sürekliliğini sağlama,
- Güncelleme ihtiyaçlarına cevap verebilme,
- Veritabanından beklenen performansı sağlama.

## 1.5.2. SON KULLANICILAR

Son kullanıcılar (end users), 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 programcılığı tarafından geliştirilen yazılımları kullanırlar.



## 1.5.3. SİSTEM ANALİSTLERİ VE UYGULAMA PROG.

---

Sistem analisti son kullanıcıların gereksinimlerini belirleyen ve standart işlemler yoluyla bu gereksinimleri karşılayabilecek ayrıntıları belirleyen kişi ya da kişilerdir.

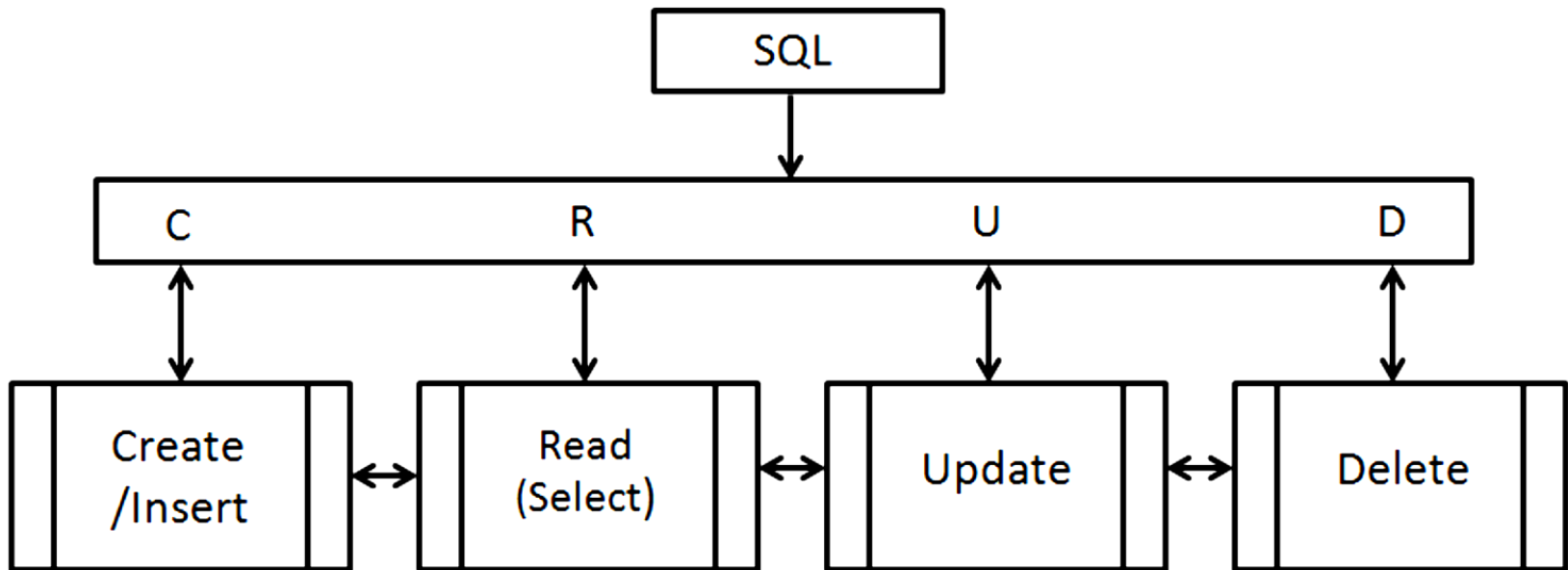
Uygulama programcıları ise sistem analisti tarafından belirlenen ayrıntıları program hâline getiren ve daha sonra test eden, hataları ayıklayan ve belgeleyen kişilerdir.

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.



## 1.6. YAPISAL SORGULA DİLİ (SQL) NEDİR?

SQL, veriyi sorgulamak, manipüle etmek ve tanımlamak ve aynı zamanda erişim kontrolü sağlamak üzere neredeyse tüm ilişkisel veritabanlarında kullanılan bir programlama dilidir.



## 1.6.1. SQL İLE ELE ALINAN KONULAR

---

CRUD işlemleri

İleri SELECT Sorgulamaları, Gruplama İşlemleri

SQL içerisinde yer alan;

- Aritmetik Operatörler, Kümeleme Operatörleri
- Matematiksel Fonksiyonlar, Karakter ve Tarih-Zaman Fonksiyonları

T-SQL,

Sanal Tablolar (Views),

Saklı Yordamlar (Stored Procedures)

Kullanıcı Tanımlı Fonksiyonlar.

Tetikleyiciler (Triggers) ve İşlemler (Transactions)

## 1.7. VERİTABANI OLUŞTURMA AŞAMALARI

---

Bir veritabanı oluşturulurken çeşitli aşamalardan geçmektedir. Bu aşamaları aşağıdaki gibi sıralayabiliriz.

Bir veritabanının kavramsal modellenmesi;

- + Kavramsal modelleme (sistemin analizi) yapılması,
- + Varlık–ilişki modelinin oluşturulması,
- + İlişkisel modelin oluşturulması.

Veritabanının mantıksal kurulumunun yapılması;

- + Kavramsal modellerin VYTS bileşenlerine dönüştürülmesidir.

Veritabanının fiziksel olarak kurulması;

- + Veritabanının programlanması safhasıdır.

## 1.7. VERİTABANI OLUŞTURMA AŞAMALARI (DEVAM...)

---

- Sistem analizi aşamasında, veritabanının tüm bileşenlerinin çok detaylı bir tanımlanması yapılmaktadır. Yani bu sistemde nelerin gerekli olduğu ayrıntılı bir şekilde incelenmektedir. Bu inceleme kod tabanlı değil metin tabanlı bir incelemedir.
- Burada ihtiyaçlar belirlenirken son kullanıcının görüşlerine yer verilmektedir. Yani sistemde hangi verilerin tutulacağı ve ne türden işlemlerin yapılacağı sistemi bizzat kullanacak kullanıcıyı da sorularak hazırlanabilmektedir.

## 1.7. VERİTABANI OLUŞTURMA AŞAMALARI (DEVAM...)

---

- Kavramsal modelleme ise veritabanı kurulum aşamalarından en önemli olandır. Sistem analizi yapıldıktan sonra elde edilen bulgular kullanılarak kavramsal bir model oluşturulmaktadır.
- Kavramsal modelleme aslında bir nevi bir programın analizidir. Eğer kavramsal modelleme doğru yapılamazsa ardından gelen fiziksel modelde de ciddi problemler ortaya çıkabilmektedir.
- Burada veritabanının kavramsal bir modeli oluşturulmaktadır. Bu model oluşturulurken de varlık – ilişki modeli kullanılmaktadır. Kavramsal modeli bir sonraki slaytlarda ayrıntılı olarak inceleyeceğiz.

## 1.7.1. KAVRAMSAL MODELLEME

---

- Kavramsal model, alanında uzman olmayan kişiler tarafından bile kolaylıkla anlaşılabilen bir modeldir.
- Kavramsal model, ilgili uygulama alanına ait kavramların tanımlandığı, bu kavramlar arasındaki ilişkilerin doğal dil cümleleriyle ifade edildiği, çok kolay anlaşılabilir ve çeşitli uygulamalarla uyumlu bir modeldir.
- Ayrıca, bir uygulama alanı hakkında elde edilen bilgiler çoğunlukla o alana ait iş kurallarını da tanımlamaktadır.

## 1.7.1. KAVRAMSAL MODELLEME (DEVAM...)

---

- Dolayısıyla, herhangi bir uygulama alanına ait kavramsal modelin yaratılması ve gerçekleştirilmesi veritabanı modeli geliştirilecek olan uygulama için en üst seviyede model geçerliliğinin de sağlanması anlamına gelmektedir.
- Bir başka ifade ile de bu model, veri tabanında tutulacak verilerin yüksek seviyede bir gösterimini ve açıklamasını vermektedir.

## 1.7.1. KAVRAMSAL MODELLEME (DEVAM...)

- Kavramsal model, fiziksel model ile karşılaştırıldığında;
  - ✓ fiziksel model, kullanılmak istenen VTYS üzerinde verilerin bilgisayarda tutulması, kayıtların düzenlenmesi ve erişim yöntemlerinin belirlenmesi vb. işleri gerçekleştirirken,
  - ✓ kavramsal model VTYS ve kısıtlamalardan bağımsız genel amaçlı bir model ortaya koymayı amaçlamaktadır.
- Kavramsal model herhangi bir sistem üzerinde değil kâğıt üzerinde veya bir tasarım yazılımı kullanılarak yapılmaktadır. Yani sistem analizi yapılmaktadır ve bu genellikle sözel bir bölümdür. Problemin detaylı tanımlaması yapılmaktadır.



## 1.7.1. KAVRAMSAL MODELLEME (DEVAM...)

---

- Şimdi kavramsal modele bir örnek verelim.
- “ ..... Üniversitesi Kütüphanesinde kullanılmak üzere bir veritabanı modellenmek istenmektedir. Kütüphanede kitaplar içeriklerine göre tasnif edilmektedir. Her kitaba ait bir kitap numarası ve kitap adı vardır. Kitaplar içeriklerine göre tasnif edildiğinden birde raf numarası verilmektedir. Kütüphane görevlilerinin kendilerine ait kullanıcı adı ve şifreleri vardır. Kütüphane görevlisi ödünç kitap verme, kitap iadesi alma, kitap ekleme ve çıkarma haklarına sahiptir.”

Devamı sonraki slaytta...

## 1.7.1. KAVRAMSAL MODELLEME (DEVAM...)

---

- “Kütüphaneden ödünç kitap öğrencilere, öğretim elemanlarına ve diğer personele verilebilmektedir. Kitaplar öğrencilere ödünç verilecekse öğrenci numarası, adı, soyadı, fakülte adı ve bölümü bilgilerine, personele verilecekse personel numarası, adı, soyadı, fakülte adı ve bölümü bilgisine, diğer personele verilecekse, personel numarası, adı, soyadı ve çalıştığı birim bilgilerine ihtiyaç vardır. Kitabın ödünç verilme süresi 15 gündür. Zamanında getirilmeyen kitaplar için personele gün başına ceza kesilir.”

# LABORATUVAR ÇALIŞMASI...

---

- Bir öğrenci yurdu için yazılım geliştirilecektir. Bu sürecin kavramsal modelini oluşturunuz.

# *TEŞEKKÜRLER...*

## *SORULARINIZ ?*

---