# VERİ TABANI ÖZET

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 aracılıyla 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. Günümüzde ise 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. Veritabanı sistemleri de kısaca veri kaynaklarının yönetiminden sorumlu sistemlerdir.

\*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 SystemBilgi Yönetim Sistemi) adıyla ilk ticari VTYS (Veritabanı Yönetim Sistemi) geliştirilmiş ve bu yapı hiyerarşik veri modeline temel teşkil 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.

Günümüzde sıklıkla karşılaşılan bir kavram hâline gelen veritabanı çok genel anlamda, bir kurumun ihtiyaç duyduğu ve kullandığı veriler bütününü 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 hâlleriyle karar verme konusunda da karar vericilere bir katkı sağlayamazlar.

Bilgi; yalın tanımıyla verinin işlenmiş ve karar verme sürecine destek olacak duruma dönüştürülmüş biçimidir. Söz konusu 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.

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.

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.

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 ulaşılıncaya kadar ilgili dosyadaki tüm verilerin sırayla okunması gerekir. Geçmişte kullanılan müzik kasetleri bu tür erişim kullanımına örnektir**

**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**

**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 sakıncaları**

**• Veri tekrarı ve veri tutarsızlığına yol açar.**

**• Veri paylaşımına olanak vermez.**

**• Uygulamalarda ihtiyaç duyulan değişikliklerin gerçekleştirilebilmesi için uzmanlık bilgisi gerektirir.**

**• İstenilen veriye ulaşmada güçlükler bulunur.**

**• Verilerin güvenliği ve gizliliği konusunda sorun yaşanır.**

**• Veriler ve uygulamalarla ilgili belirli bir standart yoktur.**

**• Verileri yedekleme ve kurtarma konusunda güçlükler yaşanır.**

**Veritabanı yönetim sistemlerinin üstünlükleri**

**• 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**

* **VTYS’ler, sağladıkları avantajlara rağmen maliyet açısından geleneksel dosya sistemlerine göre dezavantajlıdır.**
* **Veri bütünlüğü, veritabanında yer alan bir verinin farklı uygulamalarda kullanımı sözkonusu olduğunda veri üzerinde yapılacak bir değişimin verinin kullanıldığı diğer uygulamalara da yansıtılması anlamına gelir. 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. Veritabanı yönetim sistemlerinde ise sunucu/istemci mimarisi kullanılarak veriler tek bir merkezden (sunucu bilgiyasar), ağ üzerinde yer alan ve erişim yetkisi olan tüm kullanıcılara (istemci bilgisayarlar) aynı anda veriye erişme olanağı sunar.**
* **Kullanımda Üst Düzey Uzmanlık Gerektirmemesi, Geleneksel dosya sistemlerindeVerilerin 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 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**
* **Verilerin Gizliliğinin ve Güvenliğinin Sağlanması, Bu anlamda geleneksel dosya sistemlerinde gizlilik ve güvenliğin sağlanması pek kolay değildir. Veritabanı yönetim sistemlerinde, veritabanı tasarlanır ve oluşturulurken veritabanı kullanıcılarının kimler olacağı ve hangi yetkilere sahip olacağı da belirlenir. Kullanıcı grupları ve yetkileri ile ilgili tanımlar veritabanındaki diğer tanımlarla birlikte saklanır.**
* **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 oluşturulması ve uygulanabilmesini olanaklı kılar.**

**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, veritabanı yöneticisi ve veritabanı tasarımcısı olarak iki başlık altında incelenebilir.**

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

**Veritabanı yöneticisi ya da veritabanı tasarımcısı biçiminde ayrım 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: Veritabanın 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 (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 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.**

**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 ögeler 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.**

**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.**

**VERİTABANI YÖNETİM SİSTEMLERİNİN MİMARİSİ**

**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 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**

**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**

**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.**

**Veri Bağımsızlığı**

**Şema düzeyleri arasındaki dönüştürme kapasitesi olarak tanımlanabilen veri bağımsızlığı (data independence) yine üç şema mimarisi ile açıklanabilir**

1. **Mantıksal veri bağımsızlığı (logical data independence), kavramsal şemanın dışsal şemalarda ya da uygulama programlarında değişiklik yapılmaksızın değiştirilebilmesi anlamına gelir**
2. **Fiziksel veri bağımsızlığı (physical data independence), kavramsal şemada bir değişiklik yapılmaksızın içsel şemada değişiklik yapma kapasitesidir,**

**Veritabanı Yönetim Sistemlerinde Kullanılan Diller**

**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 pekçok 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.**

**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**

**Bu yapı, başlangıç noktası ağacın kökü, bağlantılı kılınacak noktalar dallar ve ana dallara bağlı alt dallar olarak düşünülebilen bir yapı biçimindedir.**

**Ağ Veritabanı**

**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 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.**

**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.**

**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 kullanıcılı 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 kullanılabilmektedir. 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 erişilebilmektedir.**

**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.**