Veritabanı yönetim sistemleri, sektör için kıymetli olan veriyi daha hızlı ve doğru bir şekilde ihtiyacı karşılamak için geliştirilmiş  ve geliştirilmeye devam edilen bir alandır. Bilgi, rakiplere üstünlük kurabilmek için doğru zamanda ve en hızlı bir şekilde edinilmesi gerekmektedir ve bu yüzden sektörde oluşan bu rekabet veritabanı yönetim sistemlerinin önemini ve gelişme hızını doğrudan etkilemektedir.

**Veri:** Ham gözlemler, işlenmemiş gerçekler ya da izlenimlerdir. Tek başlarına bir anlam ifade etmezler ve karar verme sürecinde etkili değillerdir.

**Bilgi:** Verilerin çeşitli yöntemlerle (kaydetme, sınıflama, analiz, hesaplama vb. ) anlam kazandırılmış ev karar vericilere karar sürecinde yardımcı olabilecek duruma getirilmiş halidir.

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

**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.**

1-Veritabanı Tanımlamak: Veritabanında kullanılacak verinin adı, tipi, uzunluğu gibi veri yapıları ve özellikleri belirlenir.

2-Veritabanı Oluşturmak: Veritabanını yaratma ve veriyi depolama sürecidir. Depolanacak verinin fiziksel bellek alanının belirlenmesi ve bu verinin bu alana aktarılmasını içerir.

3-Veritabanında işlem yapmak: Veritabanını izleme, veriler üzerinde sorgulama yapma ,veri tabanında değişiklikler yapma ve rapor elde edebilme sürecidir.

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

**1- Sıralı erişim:** Verilerin baştan sona taranarak istenilen verilere erişimini içeren yöntemdir. Örnek olarak kaset çalarlar verilebilir. Kaset çalarda 10 adet müzik varsa ve 5. müziği dinlemek istiyorsak ilk 4 müziğin dinlenilmesi ya da sarılması gerekmektedir. Yani 5. veriye ullaşmak için 1,2,3 ve 4. veriler sıralı bir şekilde atlanılmalıdır.

İkinci bir örnek olarak telefon rehberlerinde istenilen kişiyi bulabilmek için baştan aşağı rehberi incelemek ya da bu örneğe benzer bir örnek olarak yoklama listesindeki bir öğrenciyi bulabilmek için baştan aşşağı öğrencilerin isimlerine bakmak söylenilebilir.

Geleneksel Veritabanı Sisteminin sorunu neydi?

Bir kurumun ismini klasör ismi olarak hayal edersek o kuruma ait her alt departman kendi verilerini kurumun alt kümesi olacak şekilde departman kümesi içine yazacaktır. Bu veriler çakışabilir. A departmanında olan bir veri tekrardan B departmanında kullanılabilir. Bu da kurum kümesinde eleman tekrarına yani veri tekrarına sebep olur. Bu da depolama alanını gereksiz kullanmak , verilerin optimize olmaması, maaliyet gibi sorunları beraberinde getirir.

•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 sağladığı faydalar da elbette bunun tam tersidir. VTYS'lerin dezavantajı olarak ise sağladığı bu fonksiyonların getirdiği yüksek maaliyetlerdir. Gerekli donanım, yazılım ve veritabanı eğitimlerinin geliştirilmesi için yüksek maaliyetler harcanmaktadır. Güvenlik, depolama, yedekleme, kurtarma gibi fonksiyonların maaliyetleri de yüksektir. O yüzden tasarruf edebilmek için değişmesi beklenmeyen basit ve iyi tanımlanmış veritabanı uygulamaları sözkonusu olduğunda geleneksel veritabanı iyi bir tercih olabilir.

**VTYS AVANTAJLARI ( BAŞLIK HALİNDE ) :**

**1-**Gereksiz Veri Tekrarı ve Veri Tutarsızlığının Önlenmesi

**2-**Veri Bütünlüğünün Sağlanması

**3-**Veri Paylaşımının Sağlanması

**4-**Kullanımda Üst Düzey Uzmanlık Gerektirmemesi

**7-**Verilerin Gizliliğinin ve Güvenliğinin Sağlanması

**8-**Standart Yapı ve Kuralların Uygulanabilir Olması

**VTYS ve GELENEKSEL DOSYA SİSTEMİ KIYASLAMASI: Geleneksel dosya sistemleri, verilerin tekrarlanabileceği ve tekrarlanan verilerde yapılan değişikliklerin diğer verilere kolayca yansıtılamayacağı için veri güvenliğini zedeleyebilir ve kurumları yanıltabilir. Ayrıca, çakışan dosyaların depolanması gerektiğinde, veri depolama donanımı miktarını artırarak maliyeti de artırabilir. Veri güvenliği ve kullanıcıların verilere erişimi, veritabanı yönetim sistemlerinde sağlanabilirken, geleneksel dosya sistemlerinde bu daha zordur. Dolayısıyla, verilerin güvenli bir şekilde paylaşılması mümkün olmayabilir. Ayrıca, geleneksel dosya sistemlerinde iki kullanıcı aynı anda veriye erişmek isterse, ikinci kullanıcının erişim talebi, ilk kullanıcının veriye erişimini tamamlamasını beklemek zorunda kalabilir. Ancak veritabanı yönetim sistemlerinde, büyük veri mantığıyla aynı anda birden fazla kullanıcı veriye erişebilir ve bu da zamandan tasarruf sağlar.**

**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 yöneticisi ve Veritabanı tasarımcısının görevleri:**  
  
-Veritabanı Tasarımını Yapma

-Bütünlük Kısıtlamalarını Belirleyip Tanımlama

-Veritabanı Kullanım Yetkilerini Tanımlama

-Veritabanı Güvenliğini Sağlama

-Veritabanının İşletimini İzleme ve Sürekliliğini Sağlama

-Güncelleme İhtiyaçlarına Cevap Verebilme

-Veritabanından Beklenen Performansı Sağlama

Veri soyutlama, verilerin düzenlenmesini ve depolanmasını daha kolay anlaşılır hale getirme işlemidir. Bu işlem, verilerin teknik detaylarını gizleyerek, kullanıcıların verilere daha rahat erişmesini sağlar. Veri modeli, bu soyutlamaları gerçekleştiren bir araçtır. Üç temel veri modeli bulunmaktadır.

Fiziksel veri modeli, verilerin bilgisayar ortamında nasıl depolanacağını belirtir. Bilgisayar belleğinde ve depolama birimlerinde verilerin nasıl düzenleneceğini tanımlar.

Kavramsal veri modeli ise, varlıklar, öznitelikler ve ilişkiler gibi kavramları kullanarak veriyi tanımlar. Bu model, verinin mantıksal yapısını temsil eder. Ayrıca, nesneye yönelik modellemeyi de kavramsal modelleme olarak tanımlayabiliriz. Bu, gerçek dünyadaki nesnelerin ve ilişkilerin veritabanında nasıl temsil edileceğini belirler.

Veri tabanı şeması genellikle sabittir, yani sık sık değişmez. Ancak bazı durumlarda, veri tabanının yapısında değişiklikler yapılabilir. Bu değişiklikler, veri tabanının tasarımında yapılan değişiklikleri ifade eder.

Veri tabanı mimarisi, veri modelleri ile aynıdır, yani fiziksel, kavramsal ve dışsal düzey. Fiziksel düzey, verinin nasıl depolandığını ifade ederken, kavramsal düzey verinin mantıksal yapısını tanımlar. Dışsal düzey ise, kullanıcıların veri tabanını gördükleri görünümü ifade eder.

Veritabanı yönetim sistemi yazılımları, farklı veri tabanı ihtiyaçlarını karşılamak için kullanılır. Örneğin, MS SQL Server, Oracle, MySQL gibi yazılımlar yaygın olarak kullanılanlardır. Bu yazılımlar, farklı işletim sistemlerinde çalışabilir ve farklı veri tabanı türlerini desteklerler.

**1. Entity (Varlık) :** Hakkında bilgi depolamak istediğimiz bir nesne olarak düşünebiliriz.

ekran görüntüsü, yazı tipi, metin, dikdörtgen içeren bir resim

Açıklama otomatik olarak oluşturuldu

***Weak Entity (Zayıf Varlık) :*** Zayıf bir varlık, yalnızca kendi özellikleri ile benzersiz bir şekilde tanımlanamayacağı için başka bir varlığa Foreign Key ilişkisi ile tanımlanması gereken bir varlıktır. Yani kısaca, zayıf varlığın Primary Key’i yoktur. bu yüzden zayıf varlık olarak adlandırıyoruz.

yazı tipi, dikdörtgen, ekran görüntüsü, metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**2. Action (Eylem) :** İki varlığın veritabanında **nasıl bilgi paylaştığını** gösterir. Aşağıda göreceğiniz üzere eşkenar dörtgene benzer bir şekil ile temsil ediliyor.

sarı, yazı tipi, logo, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

**3. Attribute (Özellik) :** Varlıkların özelliklerini gösterir. “Key Attribute” dediğimiz özelliğin altı çizili olanı ile Primary Key özelliği gösteriyoruz.(örn. OgrenciNo)  
Elips ile temsil ediliyor.

yazı tipi, metin, daire, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

***Multivalued Attribute :*** Türkçe’ye “Birden Fazla Değerli Özellik” olarak çevirebiliriz. Adından da anlaşılacağı üzere bu özellikler birden fazla değer alabilir. Örneğin bir öğretmenin birden fazla telefon numarası olabilir. (örn. Sabit telefon, cep telefonu)

metin, yazı tipi, logo, grafik içeren bir resim

Açıklama otomatik olarak oluşturuldu

***Derived Attribute (Türetilmiş Özellik):*** Bu özellikler başka bir özelliğe dayanır. Örneğin bir öğrencinin bilgilerini tuttuğumuz tabloda **Doğum Günü** bilgisini tutuyoruz, buna ek olarak birde **Yaş**ını tutuyoruz diyelim. İşte burada yaş bilgisi Türetilmiş Özellik oluyor. Yani aslında biz doğum tarihini kullanarak öğrencinin yaşını bulabiliriz. Ama bazı durumlarda erişmek istediğimizde hızlıca erişebilmek için türetilmiş özelliği kullanabiliyoruz.

metin, yazı tipi, grafik, logo içeren bir resim

Açıklama otomatik olarak oluşturuldu

**4. Cardinality (Sayısallık):** Sayısallık, bir varlığın kaç örneğinin başka bir varlığın bir örneğiyle ilişkili olduğunu belirtir.

metin, çizgi, ekran görüntüsü, makbuz içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bire bir

metin, dikdörtgen, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

ÇOKA ÇOK

metin, iş kartı, dikdörtgen, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bire çok   
metin, yazı tipi, çizgi, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu