**VERİTABANINA İLİŞKİN TEMEL KAVRAMLAR**

Günümüzde sıklıkla karşılaşılan bir kavram hâline gelen veritabanı çok genel anlamda, bir kurumun ihtiyaç duyduğu ve kullandığı veriler bütününü ifade eder.

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.

**Geleneksel dosya sistemlerinin sakıncaları**

*•Veri tekrarı ve veri tutarsızlığına yol açar.*

*•Veri paylaşımına olanak vermez.*

*•Uygulamalarda ihtiyaç duyulan değişikliklerin gerçekleştirilebilmesi içinuzmanlık 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.*

**VERİTABANI KULLANICILARI**

*\*Veritabanı Sorumluları*

*\*Veritabanı Yöneticisi*

*\*Veritabanı Tasarımcısı*

*\*Son Kullanıcılar*

*\*Standart Kullanıcılar*

*\*Sıradan ya da Parametrik Kullanıcılar*

*\*Gelişmiş Kullanıcılar*

*\*Bağımsız Kullanıcılar*

*\*Sistem Analistleri ve Uygulama Programcıları*

**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. 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.

**VERİTABANI TÜRLERi**

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)*

**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

**

Samet KEKEC – Veri Tabanı Sistemleri Özeti.