

Bilgisayar Programcılığı  
2.Sınıf 1.Öğretim

No: 223010710109

Majed Al Ibrhim

## Veritabanı Sistemlerinin Temelleri

### Giriş

#### 1. Veritabanı Sistemlerinin Temelleri :

günümüzde üretim faktörleri içinde yer almaktır ve sermaye, hammadde, enerji ve iş gücü olarak sıralanan geleneksel üretim faktörleri kadar önem taşımaktadır. İşletmelerde verilerin toplanması ve işlenmesi için kullanılan araçlar, teknikler, yaklaşımlar geçmişle günümüz arasında farklılık gösterse de bilgi sistemlerinin taşıdığı önem aynıdır.

#### 2. Veritabanının Tarihçesi

Bir konuya 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 almamasına neden olmuştur. 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 istenilen.

#### 3. Veritabanı Sistemleri

Kütüphanelerde herhangi bir kitaba erişmek için kullanılan kitap künhe bilgisi bu tür düzenlemelere örnek olarak verilebilir. Bu nedenle günümüzde veritabanı denilince akıllara bilgisayar ortamında oluşturulan veritabanları gelmektedir.

İzleyen kesimde 1960'lı yıllarda itibaren veritabanı uygulamalarının tarihsel gelişimine ilişkin kısa bilgilere yer verilmiştir. DBTG yaptığı çalışmalar sonucunda ağ veri modelleri için dil özelliklerini, veritabanı şemalarını, veri tanımlama ve veri işleme dili gibi önemli veritabanı bileşenlerini tanımlamıştır.

Veritabanı Yönetim Sistemi bu dönemde bilinen bir terim hâline gelmiştir.

#### 4. VERİTABANINA İLİŞKİN TEMEL KAVRAMLAR

Günümüzde sıkılıkla karşılaşılan bir kavram hâline gelen veritabanı çok genel anlamda, bir kurumun ihtiyaç duyduğu ve kullandığı veriler bütünü ifade eder. Bununla birlikte her düzenli veri topluluğunu veritabanı olarak tanımlamak da doğru değildir. Veritabanı herhangi bir kurumda birden fazla uygulamada ortak olarak kullanılabilen verilerden oluşur. Veritabanında saklanan veriler durağan nitelikte değişmez veriler değildir. Ekleme, silme ya da güncelleme işlemleri ile veritabanındaki veriler değiştirilebilir .

#### 5. VERİTABANINA İLİŞKİN TEMEL KAVRAMLAR

Günümüzde sıkılıkla karşılaşılan bir kavram hâline gelen veritabanı çok genel anlamda, bir kurumun ihtiyaç duyduğu ve kullandığı veriler bütünü 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. 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, herhangi bir konuda birbiriley ilişkili olan ve amaca uygun olarak düzenlenmiş, mantıksal ve fizikal olarak tanımlanmış veriler bütündür. • 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 tekrarlanmasıne izin vermeden çok amaçlı kullanılmasına olanak verir.

#### 6. GELENEKSEL DOSYA SİSTEMLERİ VE VERİTABANI YÖNETİM SİSTEMLERİ

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. Bunun dışında ticari lisans seçeneği de mevcuttur. Özellikle web ortamında yaygın olarak kullanılan MySQL'in kullanımı kolay kullanıcı yetkilendirme sistemi güçlündür. Python'dan Java'ya kadar birçok programlama dili ile erişilebilmektedir.

**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 işletim sistemlerinde çalışır.

Verilerin belirli bir düzen içinde saklanması ve ihtiyaç duyulduğunda erişilebilmesi amacıyla bilgisayar kullanımında geçmişten günümüze farklı yaklaşım benimsenmiştir. Veritabanı yaklaşımından önce kullanılan ve geleneksel dosya sistemi olarak ifade edilen yaklaşımda veriler bilgisayarda ayrı ayrı dosyalar biçiminde saklanmaktadır.

Birbiriyle ilişkili olan veriler bir dosyada, başka bir açıdan birbirile ilişkili veriler de başka bir dosyada yer almaktadır.

## **7. Gereksiz Veri Tekrarı ve Veri Tutarlılığının Önlenmesi**

Önceki kesimde de belirtildiği gibi geleneksel dosya sistemlerinde her alt sistem ya da uygulama için gerekli olan veriler ayrı dosyalar biçiminde düzenlenmektedir. Bu durum aynı zamanda veri derleme işleminin de tekrarlı olmasına ve veri derleme için yapılan harcamaların artmasına neden olabilmektedir. Veri tekrarı aynı zamanda tekrarlanan verilerin farklı dosyalardaki değerlerinin de farklı olabileceğine ve dolayısıyla veri tutarlılığı ile uygulamalarda sorunlar yaşanmasına neden olabilir. Örneğin, bir işletmede yer alan ürünlerle ilişkin stok numarası bazı dosyalarda stok kodu olarak nitelendirilmiş olabilir.

## **8. 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.

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 gereklidir. Bu bankadaki bir hesapta 200.000 TL bulunduğu ve bu hesaptan para çekmeye yetkili iki kişinin olduğunu düşünelim. Parayı çekmeye yetkili kişilerin her ikisi de aynı anda farklı şubelerden 100.000 ve 150.000 TL çekmeye çalışın.

## **9. Kullanımda Üst Düzey Uzmanlık Gerektirmemesi**

Günümüzde kurumlar, bilgi sistemlerinden yoğun bir biçimde yararlanmaktadır ve bu bilgi sistemleri kapsamında birbiriyle ilişkili çok çeşitli ve büyük miktarda veriler üzerinde her türlü güncelleme, sorgulama, raporlama vb. işlemler yapmak istemektedir. Bu durum sistemin kullanımını önemli derecede sınırlarıdır. Çünkü uygulama programları yalnızca programın geliştirilmesi sırasında tanımlanmış olan genel ihtiyaçları karşılayabilir. Sonuç olarak bilgi sisteminin kullanımı sınırlı kalır. Bu noktada VTSY'lerin bu olumsuzlukları gideren iki yönü bulunmaktadır.

## **10. Verilerin Gizliliğinin ve Güvenliğinin Sağlanması**

Verilere ve verilerin saklandığı yapılara erişim ve erişim sonrası çeşitli düzeylerde kullanma yetkisinin tanımlanması verilerin gizliliği ve güvenliğini sağlar. Bu anlamda geleneksel dosya sistemlerinde gizlilik ve güvenliğin sağlanması pek kolay değildir.

## **11. Standart Yapı ve Kuralların Uygulanabilir Olması**

VTSY'lerin yapısal özellikleri ve veritabanı sorumlusunun varlığı veriler ve veritabanı üzerinde belirli standartların oluşturulması ve uygulanabilmesini olanaklı kılar. Standartların varlığı, veritabanı yapısını anlama ve kullanma açısından büyük kolaylık sağlamanın yanı sıra farklı sistemler arasında veri alışverişi için de çok önemli bir ihtiyaçtır.

## **12. 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 Programcısı

## **13. Veritabanı Sorumluları**

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

## **14. Veritabanı Tasarımcısı**

Veritabanı tasarımcısı veritabanında saklanacak olan verilerin tanımlanmasından ve bu verilerin depolanması ve gösterilmesi için gerekli olan uygun yapıların seçilmesinden sorumludur. Bu görevler çoğunlukla verilerin veritabanına depolanmasından ve veritabanı uygulamalarından önce yerine getirilir.

- 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 olduğunu belirlenmesi.

## **15. Son Kullanıcılar**

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. Son kullanıcılar da kendi içinde gruplandırılabilir:

Standart son kullanıcılar: Veritabanına nadiren erişim yapan fakat her seferinde farklı bilgi ihtiyacı olabilen kullanıcılardır.

- Havaalanları ve oteller için rezervasyon yapan acentaların ya da araç kiralama şir ketlerinin, gelen müşteri taleplerini cevaplayabilmek amacıyla sistemlerinde uygun yer ya da araç olup olmadığını sorgulamaları ve buna göre rezervasyon yapmaları sıradan son kullanıcılar 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 kullanıcılar 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.

## **16. VERİTABANI YÖNETİM SİSTEMLERİNİN MİMARİS**

VTYS'lerinin mimarisini geçmişten günümüze incelendiğinde, ilk veritabanı sistemlerinde, VTYS'nin tüm yazılım paketlerinin tek bir sisteme entegre edildiği, modern VTYS'lerinde ise istemci/sunucu mimarisile modüler bir yapılanmanın sözkonusu olduğu görülmektedir. Burada sözü edilen sunucu bilgisayarlar; web sunucular, veritabanı sunucuları, dosya sunucuları, uygulama sunucuları vb. olarak örneklenebilir.

İstemci modülü , VTYS'nin herhangi bir kullanıcı iş istasyonunda ya da kişisel bilgisayar üzerinde çalışan parçasıdır.VTYS'lerinin mimarisini geçmişten günümüze incelendiğinde, ilk veritabanı sistemlerinde, VTYS'nin tüm yazılım paketlerinin tek bir sisteme entegre edildiği, modern VTYS'lerinde ise istemci/sunucu mimarisile modüler bir yapılanmanın sözkonusu olduğu görülmektedir. Burada sözü edilen sunucu bilgisayarlar; web sunucular, veritabanı sunucuları, dosya sunucuları, uygulama sunucuları vb. olarak örneklenebilir.

İstemci modülü , VTYS'nin herhangi bir kullanıcı iş istasyonunda ya da kişisel bilgisayar üzerinde çalışan parçasıdır.

## **17. Veri Modelleri**

Veritabanı yaklaşımının temel karakteristiklerinden biri, veritabanının bazı veri soyutlama düzeyleri sağlamasıdır. Veri modeli , bir veritabanının mantıksal yapısını tanımlamada kullanılacak kavramlar, işlemler ve kurallar bütündür. Veri modelinin bu temel işlevlerine ek olarak veritabanının dinamik hareketlerini belirleyen kavramları içermesi de yaygınlaşmıştır.

## **18. Veri Modellerinin Sınıflandırılması**

Yüksek düzeyli ya da kavramsal veri modelleri , kullanıcıların veri algılama biçimyle ilişkili kavramları kapsar.

Düşük düzeyli ya da fizikal veri modelleri , verinin bilgisayar ortamında nasıl depolanacağına ilişkin ayrıntıları tanımlayan kavramları kapsar.

Fizikal 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 olarak adlandırılır.

veritabanı üzerinde ekleme, silme, değiştirme, veriyi geri çağırma gibi genel işlemleri içerir.

Nesneye yönelik modeller kavramsal veri modeliyle yakın ilişkili olan yüksek düzeyli uygulama modellerinin yeni üyesi olarak düşünülebilirler.

## **19.Şemalar, Örnekler ve Veritabanının Durumu**

Veritabanının herhangi bir veri modeliyle tanımlanması veritabanı şeması olarak adlandırılır. Diğer yönler diyagramdan belirlenemez. Örneğin, her bir veri kalemine ilişkin veri tipi, çeşitli dosyalar arasındaki ilişkiler ya da verilere ilişkin kısıtlar diyagramından izlenemez. Bu aynı zamanda veritabanının mevcut örnek kümesi olarak da adlandırılır. Veritabanında her şema yapısı kendi oluşum kümesine sahiptir.

## **20. Üç Şema Mimarisi**

Veritabanı yaklaşımının önemli karakteristikleri aşağıdaki biçimde sıralanabilir

Veritabanı tanımlarını depolamak için kendi kendine tanım yapabilmeyi sağlayan bir katalog kullanır.

Veri ve program izolasyonunu sağlar

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

Üç şema mimarisinin amacı kullanıcı uygulamalarını fiziksel veritabanından ayırt etmektir.

İçsel 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.

## **21. Veri Bağımsızlığı**

Şema düzeyleri arasındaki dönüştürme kapasitesi olarak tanımlanabilen veri bağımsızlığı yine üç şema mimarisi ile açıklanabilir.

Fiziksel veri bağımsızlığı , kavramsal şemada bir değişiklik yapılmaksızın içsel şemada değişiklik yapma kapasitesidir.

Veritabanına ek erişim yapısının oluşturulması, veritabanının bilgiyi çağırma performansının geliştirilmesi ya da veritabanının güncellenmesi içsel şema değişikliklerine verilebilecek örneklerdir.

Çok düzeyli VTSY'leri söz konusu olduğunda bu VTSY'lerin, farklı düzeyler arasındaki veri ve istek eşleştirme melerini haritalayan bir kataloğu da barındırması gereklidir.

## **22. Veritabanı Yönetim Sistemlerinde Kullanılan Diller**

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

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 .

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

## **23. 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 bulunur. 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. 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.

## **24. VERİTABANI TÜRLERİ**

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

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

## **25. Hiyerarşik Veritabanı**

Hiyerarşik veritabanı, en eski veri modeli olan hiyerarşik veri modelini temel alıp 1960 ve

1970'li yıllarda yaygın olarak kullanılmıştır. Bu biçimde en çok kullanılan yazılım IBM firmasının geliştirdiği IMS dir.

## **26. 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.

## **27. İlişkisel Veritabanı**

Bu tür veritabanı ilişkisel veri modelini temel almış ve ilk olarak 1970 yılında ortaya atılmıştır. Bu yapıda ilk iki veri modelinden farklı olarak birden çok ilişki biçimini kullanılabılır.

## **28. Nesneye Yönelik Veritabanı**

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.

## **29. 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. Bunun dışında ticari lisans seçeneği de mevcuttur. Özellikle web ortamında yaygın olarak kullanılan MySQL'in kullanımı kolay kullanıcı yetkilendirme sistemi güçlündür. Python'dan Java'ya kadar birçok programlama dili ile erişilebilmektedir.

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 işletim sistemlerinde çalışır.

---