**Bilgisayar Programcılığı**

**2.Sınıf 1.öğretim**

**No : 223010710051**

**Fatma Özbek**

**Veri Tabanına İlişkin Temel Kavramlar**

Veri, ham gözlemler ve izlenimler olup tek başına bir anlam ifade etmez. Bilgi ise verinin işlenmiş halidir. Veri tabanı ise birbiriyle ilişkili olan düzenlenip tamamlanmış veriler bütünüdür. Veri tabanının özellikleri;

* Veri tabanı bir kurumda birden fazla uygulamada ortak olarak kullanılabilen verilerden oluşur.
* Veritabanında sürekli niteliği olan veriler bulunur
* Veritabanı, ortak kullanılan verilerin tekrarlanmasına izin vermez. Çok amaçlı kullanılabilir.
* Veritabanında saklanan veriler durağan nitelikte değişmez veriler değildir. Ekleme, silme veya güncelleme işlemleri ile veri tabanındaki 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. 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. Bir örnek olarak öğrenci bilgilerinin yer aldığı bir dosyada, dosyanın en sonunda yer alan bir öğrenci bilgisine erişmek için dosyanın tamamının okunması gerekecektir. **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.

**GELENEKSEL DOSYA SİSTEMLERİ VE VERİTABANI YÖNETİM SİSTEMLERİ**

Verilerin saklanması ve istenildiğinde kolayca ulaşmak için günümüze kadar birçok yaklaşım benimsenmiştir. Geleneksel dosya sistemi olarak ifade edilen yaklaşımda, veriler bilgisayarda ayrı ayrı dosyalar biçiminde saklanıyordu. Geleneksel dosya sisteminde, tüm verilerin saklandığı bir ana dosyanın yanı sıra her alt sisteme özgü ve ana dosyanın altkümesi olarak düşünülebilecek birçok alt dosya türetilir. Bu durum da verinin gereksiz yere tekrarına ve dolayısıyla bellek kapasitesinin boş yere kullanılmasına yol açar.

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 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. 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 ile herhangi bir şekilde etkileşimde olan kişi ya da kişiler veritabanı kullanıcısı olup şu şekilde 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, veritabanının tasarlanması, oluşturulması ve veritabanının 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.

Veritabanı yöneticisinin veritabanına erişim yetkilerini belirleme, veritabanı kullanımının düzenlenmesi ve izlenmesini sağlama, ihtiyaç duyulan yazılım ve donanım kaynaklarını edinme biçiminde sıralanan sorumlulukları vardır.

Veritabanı tasarımcısı 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, 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.

**Son Kullanıcılar**

Son kullanıcılar, yaptıkları işler gereği veritabanına sorgulama ya da güncelleme yapmak veya rapor türetmek için erişen kullanıcılardır.

**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.

**Gelişmiş son kullanıcılar**: VTYS’nin sağladığı özellikler yardımıyla ayrıntılı olarak belirledikleri karmaşık gereksinimlerini karşılamak amacıyla veritabanını kullanan gruptur.

**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.

**Sistem Analistleri ve Uygulama Programcılar**ı

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, belgeleyen ve kaydedilmiş işlemler olarak sürekliliğini sağlayan kişilerdir.

**Veri Modelleri**

Veritabanı yaklaşımının temel karakteristiklerinden biri, veritabanının bazı veri soyutlama düzeyleri sağlamasıdır. Veri soyutlama 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, 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 , kullanıcıların veri algılama biçimiyle ilişkili kavramları kapsar. Düşük düzeyli ya da fiziksel veri modelleri, 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 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.

**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:

**a**. İçsel (fiziksel) düzey, veritabanının fiziksel depolama yapısını tanımlayan içsel şemayı içerir. İçsel şema, veriyi depolama ayrıntılarının tamamını ve veritabanına erişim yollarını tanımlayan fiziksel veri modelini kullanır.

**b**. 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.

**c.** Dışsal (görünüm) düzey, 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.

**Veri Bağımsızlığı**

Veri bağımsızlığı iki başlıkta ele alınır:

1. Mantıksal veri bağımsızlığı, kavramsal şemanın dışsal şemalarda ya da uygulama programlarında değişiklik yapılmaksızın değiştirilebilmesi anlamına gelir.

2. Fiziksel veri bağımsızlığı, kavramsal şemada bir değişiklik yapılmaksızın içsel şemada değişiklik yapma kapasitesidir. Buna bağlı olarak içsel şemada yapılan değişiklik dışsal şemalarda da bir değişiklik yapılmasını gerektirmez.

**Veritabanı Yönetim Sistemlerinde Kullanılan Diller**

VTYS’de iki şemayı tanımlamak amacıyla veri tanımlama dili olarak adlandırılan bir dil, veritabanı yöneticisi ve veritabanı tasarımcısı tarafından kullanılır. İki şema bu şekilde tanımlandıktan sonra söz konusu tanımlar, VTYS’de veri tanımlama dili derleyicisi tarafından işlenerek VTYS kataloğunda depolanacak uygun yapılar biçimine dönüştürülür. VTYS’lerde fiziksel şemayı belirlemek içinse depolama tanımlama dili adı verilen bir başka dilden yararlanılır.Üç şemalı mimarinin tam anlamıyla kullanıldığı yapılarda görünüm tanımlama dili olarak adlandırılan üçüncü bir dile ihtiyaç duyulur. Görünüm tanımlama dili, kullanıcı görünümlerini belirlemek ve bunların kavramsal şemadaki eşleştirmelerini belirlemek için kullanılır.

VTYS işleme faaliyetlerini gerçekleştirebilmek amacıyla veri işleme dili olarak adlandırılan bir dil kullanır. Veritabanından bilgi alma amacıyla sorgulama yapmak için kullanılan veritabanı dili ise sorgulama dili olarak adlandırılır.

**VERİTABANI TÜRLERİ**

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

2. Ağ veritabanı

3. İlişkisel veritabanı

4. Nesneye yönelik veritabanı

**Hiyerarşik Veritabanı:** Bu tür veri tabanlarında kullanılan veri modelinde kayıtlar, ilişkileri temsil eden ve ağaç yapısına benzeyen kök ve dallar biçiminde hiyerarşik bir yapıda oluşturulur. Bu yapı, başlangıç noktası ağacın kökü, bağlantılı kılınacak noktalar dallar ve ana dallara bağlı alt dallar olarak düşünülebilen bir yapı biçimindedir.

**Ağ Veritabanı**

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**

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.

**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.