Veritabanı Sistemlerinin Temelleri

Günümüz işletmeleri bilgiye olan ihtiyaçlarını karşılamak için bilgi sistemlerine başvururlar. Bilgi artık geleneksel üretim faktörleri gibi önemlidir ve zamanında elde edilebilmeli ve amaca uygun olmalıdır. Bilgi sistemleri, verilerin toplanması, işlenmesi ve ilgili kişilere iletilmesi için kullanılır. Bu sistemler, teknolojik gelişmelere paralel olarak donanım, yazılım, veri kaynakları, ağ teknolojileri ve insan bileşenlerinden oluşur. Veri kaynakları, işletmenin ihtiyaç duyduğu bilginin saklandığı ve erişilebildiği bileşenlerdir.

Veritabanı sistemlerinin gelişimi, verilere hızlı ve düzenli erişim ihtiyacından kaynaklanmıştır. Geçmişte veriler fiziksel olarak kâğıt dosyalarında saklanırken, günümüzde bilgisayarlar sayesinde veritabanları oluşturulmaktadır. 1960'larda ilk genel amaçlı veritabanı yönetim sistemleri geliştirilmiş, ardından ilişkisel veritabanı modeli önem kazanmıştır. 1980'lerde SQL standart hâline gelmiş ve ilişkisel veritabanı sistemleri ticari bir başarıya dönüşmüştür. 1990'larda internetin yaygınlaşması veritabanı endüstrisini büyütmüş ve çeşitli uygulama geliştirme araçları ortaya çıkmıştır. 2000'lerde internet endüstrisindeki düşüşe rağmen veritabanı uygulamaları büyümeye devam etmiş ve lider şirketler Microsoft, IBM ve Oracle olarak belirlenmiştir.

Veritabanı, bir kurumun ihtiyaç duyduğu ve kullandığı verilerin bütününü ifade eder. Veri, ham gözlemler, işlenmemiş gerçekler ya da izlenimlerdir; bilgi ise verinin işlenmiş ve karar verme sürecine destek olacak duruma dönüştürülmüş biçimidir. Veritabanı, birbirleriyle ilişkili ve amaca uygun olarak düzenlenmiş verilerin mantıksal ve fiziksel olarak tanımlandığı bir bütündür.

Veritabanı Yönetim Sistemi (VTYS), veritabanının tanımlanması, oluşturulması, işlem yapılması, kullanıcı yetkilerinin belirlenmesi, bakımı ve yedeklenmesi gibi işlemleri gerçekleştirmek için geliştirilmiş programlar bütünüdür. VTYS, verinin tanımlanması, depolanması, sorgulanması, güncellenmesi gibi işlevleri sağlar ve veritabanı sistemini oluşturur.

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.

Veriye erişimde sıralı erişim ve doğrudan erişim olmak üzere iki temel yaklaşım vardır. Sıralı erişimde istenilen veriye ulaşabilmek için tüm verilerin sırayla okunması gerekirken, doğrudan erişimde istenilen veriye doğrudan erişim mümkündür. Bu iki erişim biçimi farklı avantajlar ve dezavantajlar sunar.

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.

Zaman içinde artan veri miktarı, bu verinin depolanması için gerekli olan kapasite, veriye erişim ve işleme hızında yaşanan sıkıntılar geleneksel dosya sisteminin temel sınırlılıkları olarak ortaya çıkmıştır. Dolayısıyla bu sıkıntılar modern veritabanı sistemlerinin geliştirilmesini de gerekli hâle getirmiştir.

Veritabanı yönetim sistemlerinin avantajlarından birkaçı şu şekilde sıralanabilir,

Gereksiz Veri Tekrarı ve Veri Tutarsızlığının Önlenmesi

Veri Bütünlüğünün Sağlanması

Veri Paylaşımının Sağlanması

Kullanımda Üst Düzey Uzmanlık Gerektirmemesi

Verilerin Gizliliğinin ve Güvenliğinin Sağlanması

Standart Yapı ve Kuralların Uygulanabilir Olması

Veritabanı sorumluları, veritabanının tasarlanması, oluşturulması ve veritabanının işletim faaliyetlerinden birinci derecede sorumlu olan ve veritabanı üzerinde en fazla yetkiye sahip olan kullanıcılardır. Bunlar Veritabanı Yöneticisi ve Veritabanı Tasarımcısı olarak sıralanabilir.

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.

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

Veritabanı Yönetim Sistemlerinin (VTYS) mimarisi zaman içinde evrim geçirmiştir. İlk veritabanı sistemlerinde, VTYS tüm yazılım paketlerinin tek bir sisteme entegre olduğu bir yapıya sahipti. Ancak modern VTYS'lerinde ise istemci/sunucu mimarisi tercih edilmekte ve modüler bir yapılanma görülmektedir. Bu değişim, büyük merkezi ana bilgisayarların yerini, iletişim ağları aracılığıyla çeşitli sunucu bilgisayarların aldığını göstermektedir. Bu sunucu bilgisayarlar arasında web sunucuları, veritabanı sunucuları, dosya sunucuları ve uygulama sunucuları gibi çeşitli tipler bulunmaktadır.

Veritabanının herhangi bir veri modeliyle tanımlanması veritabanı şeması (database schema) olarak adlandırılır. VTYS şema yapılarını ve kısıtlarını (meta veri) veritabanı içinde tanımlar ve VTYS yazılımı ne zaman ihtiyaç duysa bu şemaya başvurur.

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.

Bu karakteristik özellikler üç şema mimarisinde gerçekleşmekte ve daha açıkça ortaya konmaktadır. Üç şema mimarisi 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.

Çoğu VTYS tam ve açık olarak bu üç düzeye ayrılmaz fakat bu üçlü şema mimarisini bir ölçüde destekler.

Veri bağımsızlığı şema düzeyleri arasındaki dönüştürme kapasitesi olarak tanımlanabilir. Veri bağımsızlığı iki başlıkta ele alınır:

1. Mantıksal veri bağımsızlığı

2. Fiziksel veri bağımsızlığı

Veritabanı Yönetim Sistemlerinde (VTYS) kullanılan diller, farklı kullanıcı gruplarına uygun bir arayüz sağlamak amacıyla çeşitlilik gösterir. Veritabanı tasarımı tamamlandıktan ve VTYS seçildikten sonra, veritabanı için kavramsal ve fiziksel şemaların belirlenmesi ve bu şemalar arasındaki eşleştirmelerin yapılması gereklidir.

Birçok VTYS'te, veri tanımlama dili (data definition language- DDL) olarak adlandırılan bir dil kullanılarak iki şema tanımlanır. Bu tanımlar VTYS kataloğunda depolanır ve VTYS tarafından işlenir. Veritabanı tanımlarının bu şekilde depolanması, yetkilendirilmiş kişiler tarafından bir kez yapılmasını, tanımların kalıcılığının sağlanmasını ve kullanıcıların bu tanımları kullanarak işlem yapmalarını sağlar.

Günümüzde, ilişkisel VTYS'lerin çoğunda depolama tanımlama dili yerine, fiziksel şema fonksiyonları ve parametrelerle depolama detayları belirlenir. Bu, veritabanı yöneticilerine depolama için verinin eşleştirilmesini kontrol etme imkânı verir.

Üç şemalı mimariyi kullanan yapılar için, görünüm tanımlama dili (view definition language- VDL) olarak adlandırılan üçüncü bir dil gerekebilir. Bu dil, kullanıcı görünümlerini belirlemek ve bunların kavramsal şemadaki eşleştirmelerini belirlemek için kullanılır. Ancak, pek çok veritabanı sistemi için SQL (Structured Query Language) hem kavramsal hem de dışsal (görünüm) şemaların her ikisini de tanımlamak için kullanılır.

Veritabanı Yönetim Sistemlerinin (VTYS) bileşen modülleri şu şekilde sıralanabilir:

**Disk Erişim Kontrolü ve Arabellek Yönetimi**: Disk üzerinde veritabanı ve VTYS kataloğunun depolandığı yerdir. İşletim sistemi, disk üzerinde okuma/yazma işlemlerini kontrol ederken, VTYS'in kendi arabellek yönetim modülü, disk erişimini programlayarak performansı artırır.

**Depolama Veri Yöneticisi**: VTYS'in daha yüksek düzey modülüdür ve veritabanının diske depolanmış bilgilerine erişimi kontrol eder. Bu bileşen, veritabanı veya kataloğun bir bölümüne erişir ve işlem yapar.

**Veritabanı Kullanıcıları ve Etkileşim Ara Yüzleri**: VTYS'e çeşitli kullanıcı gruplarının erişmesini sağlar. Bu gruplar arasında veritabanı yönetici personeli, standart kullanıcılar ve uygulama programcıları bulunur. Veritabanı yönetici personeli, veri tanımlama dili ve özel komutlar kullanarak veritabanını yönetirken, standart kullanıcılar interaktif sorgular kullanarak veritabanıyla etkileşime geçerler.

**Veri Tanımlama Dili Derleyicisi**: Veri tanımlama dilinde belirlenmiş şema tanımlarını işler ve bu tanımları VTYS kataloğuna depolar. Katalog, dosya isimleri ve boyutları, veri kalemleri ve türleri gibi bilgileri içerir.

**Sorgu İşleyici ve Eniyileme Bileşenleri**: Sorguları işleyen ve en iyi performansı sağlamak için sorguları optimize eden bileşenlerdir. Sorgu işleyici, sorguların anlamsal ve sözdizimsel çözümlemesini yapar ve sorguları işlemek için gerekli algoritmayı belirler. Sorgu eniyileyici ise sorguları optimize eder ve en uygun işlem sırasını ve indeksleri belirler. Bu bileşenler, sistem kataloğundan istatistiksel ve fiziksel bilgiler alarak sorguları işler.

Veritabanı Türleri

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.

1. Hiyerarşik veritabanı (Hierarchical database)

2. Ağ veritabanı (Network database)

3. İlişkisel veritabanı (Relational database)

4. Nesneye yönelik veritabanı (Object oriented database)

Hiyerarşik Veritabanı, en eski veri modeli olan hiyerarşik veri modelini temel alır. Bu veritabanı türünde, veriler kök ve dallar biçiminde hiyerarşik bir yapıda saklanır. Bu yapıda veriler, ağaç yapısına benzer bir şekilde düzenlenir, kök düğüm en üstte yer alır ve bağlantılı kılınacak noktalar dalları oluşturur.

Veriler arasındaki ilişkiler, hiyerarşinin üst bölümünde olan dalların alt bölümdeki dallarla bağlantılı olduğu şeklinde düşünülür. Ancak alt bölümdeki dallar, üstte kalan dallara yalnızca tek bir noktadan bağlantılı olabilirler. Bu nedenle ilişkiler, ebeveyn-çocuk ilişkisine benzetilir.

Hiyerarşik veritabanı yapısında, gereksiz veri tekrarı söz konusu olabilir ve erişimde sıkıntılar yaşanabilir. Herhangi bir veriye erişmek için arama işlemi kök düğümden başlar ve aşağı doğru devam eder. Bir dal silindiğinde, bu dala bağlı tüm alt dallar ve ilgili veriler de silinir.

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.

İlişkisel Veritabanı Bu tür veritabanı ilişkisel veri modelini temel almış ve ilk olarak 1970 yılında ortaya atılmıştır. 1970’li yılların sonunda kullanılmaya başlanmış ve 1985 yılından sonra kullanımı yaygınlaşmıştır. 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 veritabanları, günümüzdeki ve gelecekteki pek çok uygulamada sadece metin değil aynı zamanda multimedya unsurlarını da içeren verileri yönetmek için gereklidir. Bu veritabanları, grafik ve multimedya unsurlarını kullanmak için tasarlanmıştır, çünkü geleneksel ilişkisel veritabanları bu tür verileri düzenlemekte yetersiz kalır. Nesneye yönelik veri modelleri, bu eksikliği gidermek için geliştirilmiştir. Nesneye yönelik veritabanları, ilişkisel veri modellerinden farklı olarak daha karmaşık veri türleri üzerinde işlem yapabilir. Ancak, çok sayıda işlemi yürütme açısından ilişkisel veritabanlarından daha yavaş olabilirler. Bu nedenle, günümüzde hem ilişkisel hem de nesneye yönelik veri modellerini birlikte kullanan veritabanı yönetim sistemlerinin yaygınlaştığı görülmektedir.

Günümüzde yaygın olarak kullanılan veritabanı yönetim sistemi yazılımları şunlardır:

**MS SQL Server**: Microsoft firması tarafından geliştirilmiş ilişkisel veri modelini kullanan bir veritabanı yönetim sistemi yazılımıdır. Yalnızca Windows işletim sistemlerinde çalışır ve sunucu ya da istemci olarak kurulabilir.

**Oracle**: Oracle firması tarafından geliştirilmiş olan Oracle, büyük miktarda veriyi çok kullanıcılı ortamlarda saklama ve güvenli erişim sağlama amacıyla tercih edilir. Farklı işletim sistemlerinde kullanılabilir ancak maliyeti yüksektir.

**MySQL**: Açık kaynak kodlu bir veritabanı yönetim sistemi yazılımı olan MySQL, Unix, OS/2 ve Windows işletim sistemlerinde ücretsiz olarak kullanılabilir. Web ortamında yaygın olarak tercih edilir ve kullanımı kolaydır. Ticari lisans seçeneği de bulunmaktadır.

**Sybase**: SAP'a bağlı olan Sybase firması tarafından geliştirilen bu yazılım, orta ve büyük ölçekli uygulamalar için tercih edilir. Bankacılık sektörü ve kamu alanlarında sıkça kullanılır.

**PostgreSQL**: PostgreSQL, açık kaynak kodlu ve ücretsiz bir yazılımdır. Geniş özelliklere sahiptir ve hemen hemen tüm UNIX ve UNIX türevi işletim sistemlerinde çalışabilir.

**MS Access**: Microsoft Office paketi içinde yer alan bir veritabanı yönetim sistemi yazılımıdır. Küçük ölçekli masaüstü uygulamaları için kullanılır ve yalnızca Windows işletim sistemlerinde çalışır.

**DB2**: IBM tarafından geliştirilen DB2, küçük ölçekli işletmeler için maliyetli olsa da büyük ölçekli uygulamalarda tercih edilir. Windows, Linux ve Unix işletim sistemlerinde kullanılabilir.