

Veri Tabanı Sistemleri

BİR KURULUŞUN, erişimi ve kullanılması zor olan ve zaten bu nedenle de pek kullanmadığı kağıt üzerindeki bilgileri, yararlı hale getirmek için bilgisayara aktarma fikri işlevsel bir sonuç sağlamaz. Büyük miktarda verinin uygun şekilde saklanması ve istenilenlerin süratle alınıp incelenebilmesi göz önüne alındığında, bilgilerin sabit diskler yerine klasörlerle dolu dolaplarda kalması daha iyi olabilir. Veri tabanları ise, bilgisayarların bu klasörlerle dolu dolapların yerini alıp, insanlara istenilen bilgi taraması sonuçlarını daha çabuk verebilmesi için yaratılmıştır.

Veri tabanı sistemlerinin dayandığı temeller oldukça basit olmalarına karşın, kullanılan terminoloji ve bu sistemlerden gelir sağlayan şirketler tarafından mistik bir şeymiş gibi gösterilmişlerdir. Aslında bir veri tabanının tanımı oldukça basittir:

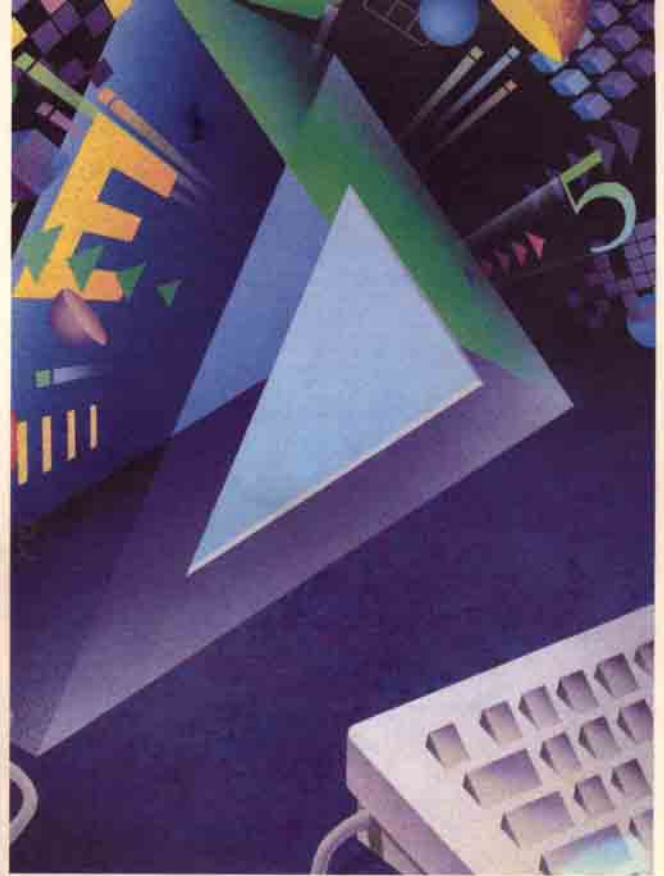
Bir veri tabanı, ulaşılabilir bir depolama ortamında düzenli bir şekilde tutulan kayıt topluluğudur. Bilgi parçacıklarının eklenmesi, değiştirilmesi ve alınmasına kontrollü izin verecek şekilde bir ya da birkaç uygulamaya hizmet eder. Veri Tabanı Yönetim Sistemi (VTYS), bu işlemleri standartlaştırılmış ve tamamiyle kontrol edilebilir şekilde yapan bir bilgisayar programları bütünüdür.

Ancak, tanımı bu kadar basit olan çok amaçlı veri tabanı sistemlerinin oluşması pek o kadar kolay olmadı. Çünkü kişiler ve kuruluşlar ihtiyaçlarının gelişeceğini göz önüne almadan, sadece o günkü bilgi işleme ihtiyaçlarını karşılayacak programları yeriniyorlardı. Çeşitli bilgisayar dilleri kullanılıyor ve bunlardan birinin yarattığı kütüğü diğeri kolay okuyamıyordu. Örneğin veri tabanı işlemleri için kullanılan ve eski bir dil olan COBOL ile yazılan kütükler Fortran veya C dili tarafından kolay kolay okunamıyordu. Bilgiye çeşitli yöntemlerle ulaşamıyordu (yani bilgiyi programda belirtilenden farklı şekilde işlemek için kendi programını

nızı yazmanız gerekiyordu) ve paylaşım olmadığından bilgiler tekrar ediliyordu (örneğin, üretim ve maaş bilgilerini içeren kütüklerin ikisinde de tüm personelin hayat hikayesinin kayıtlı olması gerekebiliyordu).

Bu durumun daha fazla devam edemeyeceğine karar veren bir grup kullanıcı, 1967'de bir araya gelerek, Database Task Group'u (Veri Tabanı Görev Grubu) oluşturdu. 1971 Nisan'ında bir VTYS için ön çalışma tamamlandı ve CODASYL (CONferencc On DATA SYtems Language: Veri Sistemleri Dili Konferansı) komitesi oluşturuldu. Görevi, veri tabanı sistemlerinin tasarımı ve işlenmesi için ortak bir dil yaratmaktır.

Günümüzde, tek kullanıcının ihtiyaçlarını karşılayabilecek basit sistemlerden, LAN ve WAN üzerinde kullanılabilen büyük sistemlere kadar birçok marka ve çeşit veri tabanı programı ve sistemi mevcuttur. Basit programlar, veri tabanını yöneten ve işleyen bir tek programdan oluşurken, veri tabanı sistemleri kullanıcı ile iletişimde bir ön yüz ve veri tabanı işlemleri için de bir sunucuya sahiptir. Bu kadar çeşitli sistemler arasında bilgi iletişiminin sağlanabilmesi amacıyla ANSI (American National Standarts Institute: Amerikan Ulusal Standartlar Enstitüsü), SQL (Structured Query Language: Yapısal Sorgu Dili) adlı VTYS'i standart olarak seçti. SQL, şu an birçok işletim sistemi ortamında destekleniyor.



VTYS'nin Yapısı

Bir VTYS'nin karmaşıklığını aşarak, tasarlanması kadar kullanımını da kolaylaştırmak amacıyla, VTYS yazılımları üç ana parçaya veya seviyeye bölünmüştür. Bu seviyeler, iç, mantiki ve dış seviyelerdir. Bu seviyelerde, veri tabanı ile işlemleri yapan programlar sırasıyla, iç şema, mantiki şema ve dış şema (alt şema) dir.

İç şema, bilginin fiziki ortamdaki (yani belirli donanım ve işletim sistemindeki) yapısı ve saklanması ile ilgilidir. Veri Saklama Tanımı Dili (VSTD) adı verilen bir alt dil ile tüm kütük işlemleri yapılır ve mantiki şemanın ihtiyaçlarının karşılanması sağlanır. Bu nedenle iç şemaya depolama şeması da denilmektedir.

Mantiki şema (genellikle sadece şema olarak da adlandırılır), çeşitli mantiki kayıtların, ilişkilerin, güvenlik ve bütünlük kontrollerinin yapıldığı kısmı içerir. VTYS'nin bir alt dili olan Veri Tanımlama Dili (VTD) ile mantiki kayıtları, veri maddelerini, ilişkileri, vb oluşturulur. Bu tabakanın fiziksel ortamdan ayrı tutulmasıyla, verilerin birçok farklı ortamda aynı şekilde işlenmesi ve iletimi sağlanmıştır.

Şemanın bir alt birimi olan dış şema (alt şema), bir uygulamanın ya da kullanıcının şemaya bakışını belirler. Uygun tanımlamalar yapıldığında, bir şirketin kayıtlarından oluşan büyük bir şema, bir muhasebe görevlisine sadece çalışanların isimlerini ve maaşlarını gösteren bir alt şema olarak gözükecektir.

Bir kullanıcı tarafından veri tabanına girilen kayıt, çeşitli kütüklerde tutulur. VTYS, bu kütüklerin yapısını, veri tabanını hazırlayan kişi tarafından belirlenmesine izin verir; böylece en verimli şekilde bir bilgiye ulaşılması sağlanabilecektir. Ulaşımın hızlı olabilmesi için bilgiler, en çok olabilecek ulaşım tipleriyle uyumlu bir şekilde kayıtlara aktarılmalıdır. Bilgilerin kayıtlara aktarılma ve oradan alınma şekli şemada tanımlandığından, programcının veri tabanını, gereken durumlarda büyük yapılanmaya yeniden ihtiyaç duymadan değiştirilebilecek şekilde hazırlaması şarttır. Bu değişikliklerin, kullanıcıları elden geldiğince az etkilemesinin sağlanması da şarttır.

Bir veri tabanı genellikle "Ad" ve ona karşılık gelen "Telefon Numarası" örneğindeki gibi tek bir kayıttan oluşmaz. Bir kütüphane örneği incelenecek olursa, okuyucu hakkındaki bilgileri içeren okuyucu kaydı, kitap konularına göre taramayı sağlayan konu kaydı, kitap kaydı ve daha birçok kayıtlar olması şarttır. Bu kayıtlar arasındaki ilişkiler çeşitli yapılarda olabilir. Günümüzde üç mantıksal yapı tipi belirlenmiştir. Bunlar, her bir kayıttan tek bir üst kayda sahip olduğu ağaç (hiyerarşik), birden fazla üst kayda sahip olunmasına izin veren ağ, ve ana bilgi maddelerinin kayıtlar arasındaki ilişkiyi sağladığı ilişkisel yapıdır. (Bunların dışında çeşitli tipler de önerilmektedir.)

Kütüphane örneğinden devam ederek, bir okuyucunun birkaç yazarı olabilen birden fazla kitap alabileceği bir

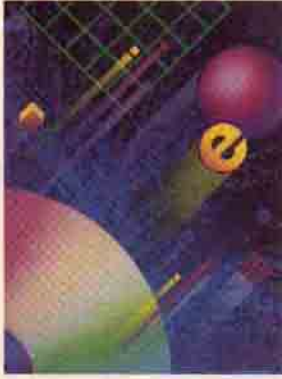
hiyerarşik sisteme göz atalım. Burada, kitap kaydının ana kaydı okuyucudur. Okuyucunun birden fazla kitabı olabilir, ancak her kitap sadece bir okuyucuda bulunabilir. Yazar kaydı da kitap kaydına aynı şekilde bağlıdır.

Ağ yapısına örnek olarak da bir üniversite ortamını ele alabiliriz. Bu veri tabanında, bölüm, teknik personel, akademik personel ve ders kayıtları bulunsun. Bu sistemde, her bir personel bir tek bölüme bağlı olsaydı ve bölümlerin dersleri ayrı ayrı olsaydı, bölüm ana kaydı altında teknik personel, akademik personel ve ders kayıtlarından oluşan bir hiyerarşik yapı ortaya çıkardı. Ancak, her ders birkaç bölüme verilebilir ve personelin bir kısmı birkaç bölüme bağlı olabilir. Bu durumda, bir ağ yapısı ortaya çıkar.

İlişkisel yapı ise, karmaşık veri tabanı sistemlerinde kullanılmak üzere hazırlanmıştır. Her bir kayıta bir birincil anahtar bulunur. Kayıtlar arası ilişkilerin tanımlandığı tablolar mevcuttur. Kütüphane örneğine geri dönersek olursak, her bir kitap kaydının birkaç yazar ve konu kaydı olması gerektiğini görürüz. Konusu ve/veya yazar aynı olan birçok kitap da olacaktır. Bu durumda konu ve yazar kayıtlarının her bir kitapla birlikte tekrar edilmesi ve bir okuyucunun kitap alması durumunda, tüm bu kayıtların taşınması oldukça zaman alır ve zor bir işlem olacaktır. Bunun yerine ilişkisel bir yapı kullanarak, uygun şekilde seçilmiş numaraların birincil anahtar olarak kullanıldığı kitap, okuyucu, konu, yazar kayıtlarının ilişki tablolarıyla birleştirilmesi daha verimlidir.

Dağıtılmış Veri Tabanı Sistemleri

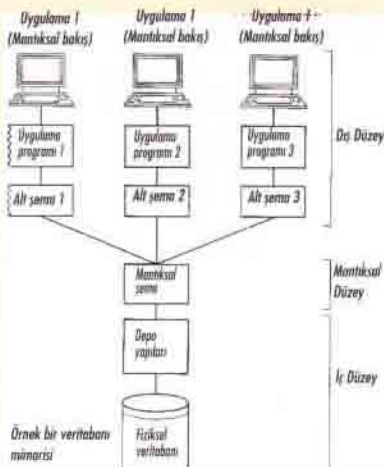
Bir bilgisayar ağı üzerindeki çoklu, mantıksal bağlar içeren veri tabanları topluluğuna dağıtılmış veri tabanı adı verilir. Büyük şirketlerden okullara kadar birçok kuruluş bu tip bir sisteme ihtiyaç duymaktadır. Örneğin bir şirket düşünelim; merkezi İstanbul'da, dağıtım merkezi Ankara'da olsun. İzmir ve Adana'da depolar, Manisa'da fabrika ve Gebze'de araştırma-geliştirme tesisleri yer alsın. Bu tip bir kuruluşun merkezleri ar-ge tesislerindeki ve depolarındaki durum, ar-ge tesisleri fabrikaların işleyişi, fabrika depodaki mal miktarı ve her bir nokta kendi personeli hakkında bilgi tutmak ve



bu bilgiye kolayca ulaşmak isteyecektir. Böylesi bir kurulumda yukarıda anlatılanlar dışında daha birçok farklı tipteki bilginin paylaşılması gerekecektir. Ve bu da Dağıtılmış Veri Tabanı Sistemlerinin (DVTS) kullanımı ile olur.

DVTS'in yaratılabilmesi için çeşitli şeffaflıklara ihtiyaç duyulmaktadır. Bir dağıtılmış VTYS'nin şeffaflığı ile, üst düzey konuların alt düzey uygulamalardan ayrılması vurgulanmaktadır. Yani, şeffaf bir sistem, çalıştığı ortama olan ilişkilerini kullanıcılarından saklayacaktır. Tamamiyle şeffaf bir dağıtılmış VTYS ile karmaşık uygulamalar yaratılabilmektedir. Veri bağımsızlığı bir VTYS'de olması gereken temel bir şeffaflıktır. Bu bağımsızlık, iki kısımdan oluşur. Mantıki veri bağımsızlığı, veri tabanının mantıksal yapısındaki değişimlerden kullanıcı uygulamalarının etkilenmemesinin sağlanmasıdır. Fiziksel veri bağımsızlığı ise, yazılan uygulamaların fiziksel yapılarıdaki değişikliklerden etkilenmemesidir. Bu, daha önce belirtilen iç şema, mantıki şema ve alt şema ayrıntısının temelinde yatan görüştür.

Bir dağıtılmış VTYS'de veri bağımsızlığına ek olarak ağ şeffaflığının da bulunması gerekmektedir. Yani, bir ağ ortamında birçok farklı işletim sistemi arasında veri aktarımı yapılırken, bunların kullanıcıdan saklanmasıdır. Kullanıcı, hem yerel, hem de ağ üzerinden yapacağı işlemleri tek bir arayüzle ve hangisinin yerel, hangisinin ağ üzerinden olduğunu bilmeden yapabilmektedir. Burada çoğaltma şeffaflığına da değinilmesi gerekir. Çoğaltma şeffaflığı, kullanılan ağıntı bant genişliğine bağlı olmadan çalışabilmesi için sıkça kullanılan bilgilerin yerel makinelerle kopyalanmasıdır. Çoğaltma işlemi sırasında tüm bir kaydın değil de, kayıtları belirli kısımlarının kopyalanması ile, daha yüksek verim elde edilebilmektedir.



Ayarlanabilirlik

Ayarlanabilirlik, bilgisayar dünyasında birçok farklı şekilde yorumlanan bir kavram haline geldi; farklı donanım ve işletim sistemlerinde bir uygulamanın çalışabilmesinden (ki bazen bunu dikey ayarlanabilirlik de denilmektedir), bir uygulamanın (ön ucun) farklı hesaplama ortamlarınca (arka uçlarda) sunulan bilgi ve servislerle ulaşabilmesine kadar değişik tanımlara sahip; dBase, SQLBase, Oracle gibi ticari sistemler dikey ayarlanabilirliğe sahip; FoxPro ve Paradox sistemleri ise sadece ön uç ayarlanabilirliği sunuyor.

Ticari olmamasına rağmen çok güçlü veri tabanı işlemleri de mevcuttur. Bunlardan en güçlüsü Postgres veri tabanı sistemidir. Bunun dışında, mbase, oqdb ve Postgres'in atası olan Ingres gibi veri tabanı sistemleri de mevcuttur. Ancak ticari olmayan ve SQL sunucusu olan tek yazılım mSQL'dir. Bu sisteme, Unix altında olduğu gibi, Windows altında da çalıştırılmaktadır.

Veri Tabanı Sistemi Hazırlanması

Bir veri tabanı sisteminin hazırlanması için bir kuruluş istediği sistemi belirterek kavramsal şemayı hazırlatmalıdır. VTYS'nin kullanımıyla kavramsal şema mantıki şema haline getirilecek, alt şemalar ve bunları çağırarak uygulama programları yaratılacaktır. Aynı zamanda işletim sistemi de hazırlanan iç şema ile makine arasındaki son bağı tamamlayacaktır.

Veri tabanlarında artık çok önemli bilgilerin saklandığı düşünülecek olursa, kullanacak veri tabanı sisteminin ne kadar dikkatlice seçilmesi gerektiği ortaya çıkacaktır. Büyük bir kuruluşun günlük işleri düzenlemek için kullanılan basit veri tabanı programlarıyla, bir dağıtılmış VTYS'nin gücünü elde etmeye çalışması ciddi bir hata olacaktır. Bunun örnekleri ve sonuçlarının sık sık görülüyor olması, oldukça düşündürüştür. Uygun bir veri tabanı uygulamasının seçimi konusunda en iyi örneklerden biri İngiliz Kütüphanesi'dir (British Library, BL).

BL, sahip olduğu 18 milyon kitabın her biri 500 sayfalık 366 ciltten oluşan kataloğunu bilgisayara aktararak OPAC (On-line Public Access Catalog) sistemini kurmaya karar verdi. Böylece kullanıcılarına daha kolay hizmet verecekti.



OPAC kurulurken dikkat çeken ilk şey, sistemin küçük parçalardan oluşturulmasının önemiydi. Bu nedenle, hem yazılımın, hem de donanımın parçalardan oluşturulmasına özen gösterildi. OPAC iki tabakalı bir sunucu-müşteri yapıya sahip. Böylece en yoğun zamanında bile, bant genişliği zorlanmıyor. Ayrıca ilişkisel VTYS yerine, tüm yazı inceleme sisteminin kullanılıyor olması da OPAC'in ilgi çekici noktalarından biri.

OPAC ön ucu, ekrana veya F tuşlarına basılmasına dayanan basit bir kullanıcı ara yüzüne sahip 486 işlemcili PC'lerden oluşuyor. Ön uç için Windows işletim sistemi seçilmiş. Bunun ilk nedeni ucuzluğu; ikincisi de çeşitli dillerdeki yazıların (Romca, Yunanca, Arapça, vb...) aynı dilde aranmasını sağlayabilmek. Ön uçtan, kitabın adına, yazarına, yayıncısına, tarihine, katalog numarasına veya beneye kadar anahtar kelimeye göre aramalar yaptırmak mümkün oluyor.

Kullanıcı sunucusu olarak Digital iş istasyonları var. Bunlar, tarama sonuçlarını bir Ingres veri tabanında saklıyorlar. Aynı zamanda yapılan işlemlerle ilgili bilgileri de incelenmek üzere tutuyorlar. Ana sunucu ise, yine Digital firmasının bir iş istasyonu.

Tüm yazı arama sistemi, dökümanların önemli kelimelerinin yazıdaki yerlerini kaydediyor. BL yeni bir katalog girdiğinde, sistem karşılaştığı tüm kelimeleri saklayarak bir ters kütük yaratıyor. Ters kütüğe bu isim, yazıdaki tüm kelimelerden yazıya yönlendiriciler yaratması nedeniyle verilmiş. Ters kütüklerin kullanımı ile veri tabanı aramaları daha hızlı olabilir. Çünkü tüm yazının arandığı kelimeleri bulmak amacıyla baştan aşağıya taranamıyorsa, sadece ters kütükteki liste inceleniyor. Veri tabanı sistemi, çok yer kaplamaması için tüm veri tabanını uygun şekilde sıkıştırılmış olarak tutuyor.

Kaynaklar
Özsu M. T. Valduriez P. Principles of Distributed Database Systems.
Yamakoudakis E.J. The Architectural Logic of Database Systems.
Pountain D. The British Library's Catalog Is On-Line, Byte, Mayıs 1995.
Richter J. Desktop Databases Pack More Power, Byte, Mayıs 1995.