**VERİTABANI SİSTEMLERİ**

Veritabanı, kurumların ihtiyaç duyduğu verilerin toplamını ifade eder. Veri, işlenmemiş gerçeklerken, bilgi işlenmiş ve karar vermeye yardımcı hale getirilmiş veridir. Veritabanı yönetim sistemi , veritabanlarını tanımlama, oluşturma, işleme, yetkilendirme ve bakım için kullanılan programlar bütünüdür. Geçmişten günümüze, veri yönetimi için sıralı erişim ve doğrudan erişim yaklaşımları benimsenmiştir. Geleneksel dosya sistemlerinde, veriler ayrı dosyalarda saklanırken, veritabanı sistemi ilişkili verilerin mantıksal ve fiziksel olarak tanımlandığı bir yapı sunar. Geleneksel dosya sistemleri, her alt sistem veya uygulama için ayrı dosyalar kullanır. Bu durum veri tekrarına, tutarsızlığa ve veri bütünlüğü sorunlarına neden olabilir. Veritabanı yönetim sistemleri, tüm verileri tek bir merkezde toplar ve veri tekrarını önler. VTSY'lerin kullanıcıları, veritabanındaki verilere basit ve etkili bir şekilde erişebilirler. Veritabanı yönetim sistemlerinde veri güvenliği ve gizliliği sağlanabilirken, geleneksel dosya sistemlerinde bu daha zordur. VTSY'ler, belirli standartların oluşturulması ve uygulanması için merkezi bir kontrol sistemi sağlar. Bu standartlar, veri yapısını anlama ve farklı sistemler arasında veri alışverişi için önemlidir. Veritabanı kullanıcıları, genellikle veritabanı sorumluları, yöneticileri, tasarımcıları, son kullanıcılar ve standart kullanıcılar gibi çeşitli sınıflara ayrılabilir. Metinde, veritabanı yöneticisinin ve tasarımcısının sorumluluklarının yanı sıra farklı veritabanı kullanıcılarının türleri ve veritabanı sistemlerinin mimarisi hakkında detaylı bilgiler bulunmaktadır. Veritabanı yöneticisinin sorumlulukları, kullanıcı yetkilerinin belirlenmesi, güvenlik sağlanması, performansın optimize edilmesi ve güncelleme ihtiyaçlarına cevap verilmesini içerirken, veritabanı tasarımcısı veri yapılarının ve ilişkilerinin belirlenmesiyle ilgilenir. Son kullanıcılar, standart, sıradan, gelişmiş ve bağımsız kullanıcılar olarak gruplandırılmıştır ve farklı gereksinimlere sahiptirler. VTYS mimarisi, istemci/sunucu mimarisi ile modüler bir yapılanmayı içerir ve veritabanının mantıksal yapısını tanımlamak için veri modelleri kullanılır. Üç şema mimarisi, içsel, kavramsal ve dışsal düzeylerde tanımlanan şemaları içerir ve VTYS'nin işlevselliğini sağlar. Veri bağımsızlığı kavramı üç şema mimarisi üzerinden açıklanmıştır. İki ana başlık altında ele alınan veri bağımsızlığı, mantıksal veri bağımsızlığı ve fiziksel veri bağımsızlığı olarak tanımlanmıştır. Mantıksal veri bağımsızlığı, kavramsal şemanın değiştirilmesini gerektirmeden dışsal şemada veya uygulama programlarında değişiklik yapabilme yeteneği olarak tanımlanırken, fiziksel veri bağımsızlığı ise kavramsal şemada yapılan değişikliklerin içsel şemayı etkilemeden yapılabilmesini ifade eder. Bu bağlamda, fiziksel veri bağımsızlığının, fiziksel ayrıntıları içeren veritabanı ve dosya sistemlerinde yaygın olduğu belirtilmiştir. Ancak, kavramsal veri bağımsızlığına ulaşmanın daha zor olduğu vurgulanmıştır, çünkü bu tür bağımsızlık yapısal değişikliklere izin verirken uygulama programlarını etkilememektedir. Metin ayrıca, veritabanı yönetim sistemlerinin (VTYS) farklı düzeylerdeki kullanıcı gruplarına uygun dil ve arayüzler sağlaması gerektiğini belirtmektedir. VTYS'nin tamamlanmasıyla veri tanımlarının yapıldığı ve VTYS kataloğuna derlendiği, bu sürecin veritabanı yaklaşımının temel özelliklerinden biri olduğu ifade edilmiştir. Ayrıca, veri işleme ve sorgulama dilleriyle ilgili bilgiler de metinde yer almaktadır. VTYS'nin veri işleme dili olarak bir dil kullandığı ve günümüzde genellikle bu dillerin ayrı olmadığı, SQL gibi geniş kapsamlı birleştirilmiş dillerin kullanıldığı belirtilmiştir. Üst kısımda çeşitli kullanıcıların ve etkileşimlerinin, alt kısımda ise verilerin depolanması ve işlenmesinden sorumlu iç bileşenlerin gösterildiği bir şema verilmiştir. Bu bileşenler arasında veritabanı yönetici personelleri, standart kullanıcılar, programcılar, ara yüzler, sorgu işleyicileri ve depolama yöneticileri gibi unsurlar yer almaktadır. Hiyerarşik veritabanı, 1960'lar ve 1970'lerde yaygın olarak kullanılan en eski veri modelidir. Bu modelde veriler, kök ve dallar biçiminde hiyerarşik bir yapıda düzenlenir. Kök, başlangıç noktasını temsil ederken, dallar ilişkileri ve alt dalları temsil eder. Bu yapıda, üst düzey dallar alt düzey dallarla bağlantılı olabilirken, alt düzey dallar yalnızca bir üst düzey dala bağlanabilir. Bu nedenle, bu model ebeveyn-çocuk ilişkisine benzetilir.

Hiyerarşik veritabanları, genellikle sunucu bilgisayarlarında çalışan yazılımlar tarafından kullanılır, örneğin IBM'in geliştirdiği IMS (Information Management System). Ancak, bu yapıda gereksiz veri tekrarı ve erişimde sıkıntılar yaşanabilir. Herhangi bir veriye erişmek için arama işlemi kök düğümden başlayarak alt düğümlere doğru devam eder. Bir dal silindiğinde, bu dala bağlı tüm alt dallar ve veriler de silinir. Ağ veritabanı, 1970'lerden 1980'lerin başına kadar popüler olan bir veritabanı türüdür. Bu tür, hiyerarşik veri modelinin sınırlamalarını gidermek için geliştirilmiş ve ağ veri modelini temel alır. Hiyerarşik modelde tek yönlü ilişkiler bulunurken, ağ modelinde düğümler arasında çoklu ilişkiler mümkündür, bu da daha karmaşık veri ilişkilerini yönetmeyi sağlar. Ağ veritabanları, özellikle birden çok ilişki türü olan verilerin etkin şekilde temsil edilmesi gerektiği durumlarda tercih edilir. Ağ veritabanları, ilişkisel veri modelini temel alır ve ilk kez 1970 yılında ortaya çıkmıştır. 1970'lerin sonundan itibaren kullanılmaya başlanan bu tür, 1985 yılından sonra daha geniş çapta kabul görmüştür. Bu veritabanı yapısında, birden çok ilişki biçimi kullanılabilir, ancak günümüzde, ilişkisel veri modeli hemen hemen tüm veritabanı yönetim sistemlerinde tercih edilen bir modeldir. Günümüzde ve gelecekteki pek çok uygulama, sadece metin bazlı verileri değil, aynı zamanda multimedya ögelerini de içeren veritabanı yönetim sistemlerine ihtiyaç duyar. Geleneksel veritabanı sistemleri, grafik ve multimedya ögelerini işlemek için uygun değildir, bu nedenle nesneye yönelik veri modelleri geliştirilmiştir. Bu modeller, multimedya içeriklerini ve farklı kaynaklardan gelen verileri bir araya getiren web uygulamalarını yönetmek için kullanılabilir. Nesneye yönelik veritabanları, ilişkisel veri modellerinden farklı olarak daha karmaşık veri türlerini işleyebilir, ancak işlem hızı açısından ilişkisel veritabanlarından daha yavaş olabilir. Bu nedenle, günümüzde hem ilişkisel hem de nesneye yönelik veri modellerini birleştiren veritabanı yönetim sistemlerinin yaygınlaştığı gözlemlenmektedir.