

Veritabanının Tarihçesi

Veritabanı olgusu, geçmişten günümüze teknolojik gelişmelerle birlikte evrim geçirmiştir. Eskiden veriler kağıt dosyalarında saklanırken günümüzde bilgisayar ortamında saklanmaktadır. Bilgisayar teknolojisinin gelişmesiyle verilerin saklanması, düzenlenmesi ve erişimi kolaylaşmıştır. 1960'lardan itibaren, Charles Bachman ve IBM gibi şirketlerin çalışmaları veritabanı yönetim sistemlerinin temellerini atmıştır. 1960'ların sonlarında COBOL dili standartlaştırılmış ve veritabanı standartları oluşturulmuştur. 1969'da Edgar F. Codd'un ilişkisel veritabanı modeli, veritabanı dünyasında devrim niteliğinde bir gelişmedir. 1970'lerde, Ingres ve System R gibi ilişkisel veritabanı sistemleri geliştirilmiştir. 1980'lerde SQL standart sorgu dili olarak belirlenmiş ve ilişkisel veritabanı sistemleri yaygınlaşmıştır. 1990'lar, internetin kullanımının artmasıyla veritabanı uygulamalarında hızlı bir büyümeye sahne olmuştur. 2000'lerde internet endüstrisinin düşüşe geçmesine rağmen veritabanı uygulamaları büyümeye devam etmiştir. Günümüzde, Microsoft, IBM ve Oracle gibi şirketlerin lider olduğu veritabanı yönetim sistemleri bulunmaktadır.

VERİTABANINA İLİŞKİN TEMEL KAVRAMLAR

Veritabanı, bir kurumun ihtiyaç duyduğu ve kullandığı verilerin düzenli bir şekilde saklandığı bütündür. Veri, işlenmemiş gözlemleri ifade ederken, bilgi, işlenmiş ve karar verme sürecine destek olacak hâle getirilmiş veridir. Veritabanı, bu verilerin düzenli ve mantıksal olarak tanımlandığı bir yapıdır. Veritabanı, sürekli nitelikte olan verileri içerir ve verilerin tekrarlanmasına izin vermez. VTYS, veritabanı oluşturma, işleme, güncelleme, ve kullanıcı yetkilerini belirleme gibi işlevleri yerine getiren bir programlar bütünüdür. Veritabanı erişimi, sıralı ve doğrudan olmak üzere iki temel biçimde gerçekleşir. Sıralı erişimde, tüm veriler sırayla okunurken, doğrudan erişimde istenilen veriye doğrudan erişim sağlanır. Bu iki erişim biçimi arasında avantajlar ve dezavantajlar bulunmaktadır.

GELENEKSEL DOSYA SİSTEMLERİ VE VERİTABANI YÖNETİM SİSTEMLERİ

Bilgisayar kullanımının ilk dönemlerinde, veriler geleneksel dosya sistemleriyle saklanıyordu. Bu sistemde, ilişkili veriler farklı dosyalarda bulunurdu ve aynı veri farklı dosyalarda tekrarlanabilirdi.

Gereksiz Veri Tekrarı ve Veri Tutarsızlığının Önlenmesi

Geleneksel dosya sistemlerinde, her alt sistem veya uygulama için ayrı dosyalar kullanılır ve bu durum veri tekrarına neden olur. Veri tekrarı, bellek kapasitesinin artmasına ve veri tutarsızlığına yol açabilir. Veritabanı yönetim sistemlerinde ise tüm veriler tek bir merkezde toplanır ve ihtiyaç duyulan her birime bu merkezden erişim sağlanır, böylece veri tekrarı önlenir.

Veri Bütünlüğünün Sağlanması

Veri bütünlüğü, bir verinin bir uygulamadan diğerine aktarılması veya silinmesi durumunda, bu değişikliğin tüm uygulamalara yansıtılması gerekliliğini ifade eder. Geleneksel dosya sistemlerinde veri bütünlüğü sağlanması zor olabilirken, veritabanı yönetim sistemlerinde kısıtlamalar ve kurallar sayesinde veri bütünlüğü kolaylıkla sağlanır.

Veri Paylaşımının Sağlanması

Geleneksel dosya sistemlerinde aynı veriye eş zamanlı erişim mümkün değildir ve erişim sıralı olarak gerçekleşir. Veritabanı yönetim sistemlerinde ise sunucu/istemci mimarisi kullanılarak birden fazla kullanıcı aynı anda veriye erişebilir. Veritabanı yönetim sistemi, verilerin ortak paylaşımını sağlar ancak eş zamanlı işlem yapmanın potansiyel sakıncaları göz önünde bulundurulmalıdır. Bu nedenle, veritabanı tasarımında belirli kısıtlamalar ve kurallarla bu olumsuzlukların önüne geçilir. Örneğin, bankacılık sektöründe, bir bankanın veritabanı tanımları arasında hesap bakiyesinin negatif olamayacağı gibi kısıtlamalar bulunabilir. Bu tür kısıtlamalar sayesinde, aynı anda yapılan işlemlerden kaynaklanabilecek olumsuz durumlar önlenir.

Kullanımda Üst Düzey Uzmanlık Gerektirmemesi

Günümüzde kurumlar, bilgi sistemlerinden yoğun bir şekilde faydalanmakta ve bu sistemler aracılığıyla birbiriyle ilişkili çeşitli ve büyük miktarda veriler üzerinde çeşitli işlemler yapmak istemektedir. Geleneksel dosya sistemlerinde veriler dosyalar biçiminde saklanırken, kurumun ihtiyaç duyduğu işlevleri gerçekleştirmek üzere ayrıca geliştirilen uygulama programlarından yararlanır. Ancak, verilerin karmaşıklığı ve uygulamaların çeşitliliği nedeniyle bu sistemler oldukça karmaşık hale gelir. Bu da uygulama programlarının geliştirilmesi için uzman kişilere olan ihtiyacı artırır ve kullanımı sınırlar.

Veritabanı Yönetim Sistemlerinin (VYS) bu olumsuzlukları gideren iki önemli yönü bulunmaktadır. İlk olarak, VYS kullanıcılarına verilerin saklanması için karmaşık fiziksel yapılarla uğraşmak yerine basit mantıksal yapıları görme imkanı sunar. İkinci olarak, VYS kullanıcılara veritabanı uygulamalarını kolayca gerçekleştirmeleri için kullanımı basit olanaklar sağlar, örneğin sorgu dili gibi. Bu da kullanıcıların önceden belirlenmiş operasyonel uygulamalarla sınırlı kalmamasını ve kullanımın yaygınlaşmasını sağlar.

Verilerin Gizliliğinin ve Güvenliğinin Sağlanması

Kurumlarda, ortak kullanılan verilerin gizliliği ve güvenliği önemlidir. Geleneksel dosya sistemlerinde bu sağlanması zordur, ancak veritabanı yönetim sistemleri kullanıcıların yetkilerini belirleyerek veri güvenliğini sağlar. Öğrenci bilgileri gibi hassas veriler için farklı kullanıcı gruplarına farklı erişim yetkileri tanımlanabilir, bu da verilerin korunmasına yardımcı olur.

Standart Yapı ve Kuralların Uygulanabilir Olması

Geleneksel dosya sistemlerinde farklı alt sistemlerin kendi uygulamalarını ve dosya yapılarını kullanmasıyla standart olmayan bir dosya yapılanması ortaya çıkar. Ancak, Veritabanı Yönetim Sistemleri (VTSY'ler) merkezi bir kontrol sistemi sunar. Bu sayede belirli standartlar veri yapısı, adlandırma, belgelendirme gibi konularda oluşturulabilir ve uygulanabilir. Bu standartlar, veritabanı yapısını anlamayı kolaylaştırır ve farklı sistemler arasında veri alışverişini sağlar.

VERİTABANI KULLANICILARI

- 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

Veritabanı sorumluları, veritabanının tasarlanması, oluşturulması ve işletilmesinden sorumlu en yetkili kullanıcılardır. Bu sorumluluk, genellikle veritabanı yöneticisi ve veritabanı tasarımcısı olarak iki ana başlık altında incelenir. Ancak, bu roller aynı kişi ya da kişiler tarafından üstlenilebileceği gibi, kurumun büyüklüğü ve kullanıcı sayısı gibi faktörlere bağlı olarak farklı kişiler tarafından da yürütülebilir.

Veritabanı Yöneticisi

Veritabanı yöneticisi, veritabanına erişim yetkilerini belirleme, kullanımın düzenlenmesi ve izlenmesini sağlama, yazılım ve donanım kaynaklarını sağlama gibi sorumlulukları üstlenir. Ayrıca güvenlik ihlalleri ve sistem yanıt süresi gibi sorunların çözümünden de sorumludur. Büyük işletmelerde bu sorumluluklar için yardımcı personele de ihtiyaç duyulabilir.

Veritabanı Tasarımcısı

Veritabanı tasarımcısı, verilerin tanımlanması ve depolanması için uygun yapıların seçilmesinden sorumludur. Potansiyel kullanıcı gruplarıyla etkileşimde bulunarak veritabanı tasarımını oluşturur ve her kullanıcı grubuna özel görünüm geliştirir. Temel görevleri arasında veritabanı tasarımı, bütünlük kısıtlamalarının belirlenmesi, kullanıcı yetkilerinin tanımlanması, güvenliğin sağlanması, işletim izleme ve sürekliliğin sağlanması, güncelleme ihtiyaçlarına cevap verme ve performansı sağlama yer alır. Bu görevler, veritabanının etkili bir şekilde yönetilmesini ve kullanıcı ihtiyaçlarını karşılamasını sağlar.

Son Kullanıcılar

Son kullanıcılar, veritabanına sorgulama, güncelleme yapma veya rapor oluşturma gibi işler için erişen kullanıcılardır. Bu kullanıcılar, uygulama programcıları tarafından geliştirilen yazılımları

kullanarak veritabanıyla etkileşime geçerler. Son kullanıcılar, standart, sıradan, gelişmiş ve bağımsız olmak üzere dört gruba ayrılabilirler:

Standart son kullanıcılar: Nadiren veritabanına erişen ve her seferinde farklı bilgi ihtiyacı olan kullanıcılardır. Genellikle gelişmiş veritabanı sorgu dillerini kullanarak isteklerini belirtirler. Bu grup, genellikle üst düzey yöneticileri içerir.

Sıradan son kullanıcılar: Veritabanı üzerinde sürekli sorgulama ve güncelleme yapması gereken kullanıcılar. Temel iş fonksiyonları, standart sorgu ve güncelleme işlemlerini içerir. Banka memurları, rezervasyon yapmak isteyen acentalar gibi örnekler verilebilir.

Gelişmiş son kullanıcılar: Karmaşık gereksinimlerini karşılamak için veritabanını detaylı olarak kullanan grup. Mühendisler, bilim adamları, işletme analistleri gibi kullanıcılar bu gruba dahildir.

Bağımsız son kullanıcılar: Hazır paket programlarını kullanarak kişisel veritabanılarını yöneten kullanıcılar. Vergi paketi kullanıcıları gibi örnekler verilebilir.

Her kullanıcı grubu, veritabanı yönetim sistemlerinin (VTYS) sağladığı özellikleri kullanır. Standart son kullanıcılar genellikle sadece kullanıcı ara yüzlerini ve temel işlemleri bilirken, gelişmiş son kullanıcılar karmaşık gereksinimlerini karşılamak için VTYS'nin pek çok özelliğini öğrenmeye çalışır. Bağımsız son kullanıcılar ise belirli yazılım paketlerini kullanarak ihtiyaçlarını karşılar.

Sistem Analistleri ve Uygulama Programcıları

Sistem analistleri, özellikle sıradan son kullanıcıların ihtiyaçlarını belirleyen ve bu gereksinimleri standart işlemlerle karşılayabilecek detayları belirleyen kişilerdir. Uygulama programcıları ise sistem analistlerinin belirlediği detayları programlayan, test eden, hataları gideren, belgeleyen ve işlemleri sürekli hale getiren kişilerdir. Genellikle yazılım geliştiriciler veya yazılım mühendisleri olarak da adlandırılan bu kişilerin, VTYS'nin sağladığı tüm özellikleri bilmeleri ve kullanmaları gerekmektedir.

VERİTABANI YÖNETİM SİSTEMLERİNİN MİMARİSİ

VTYS'lerinin evrimi incelendiğinde, ilk sistemlerde tüm yazılım paketlerinin tek bir sistemde entegre olduğu, modern VTYS'lerinde ise istemci/sunucu mimarisi ve modüler yapıların kullanıldığı görülür. Büyük merkezi ana bilgisayarlar yerini, iletişim ağları aracılığıyla çeşitli sunucu bilgisayarlarına bırakmıştır. Basit bir istemci/sunucu mimarisinde sistem iki modüle ayrılır: İstemci modülü, kullanıcı iş istasyonunda veya kişisel bilgisayar üzerinde çalışır ve veritabanına erişimi sağlar; Sunucu modülü ise veri deposuyla iletişim kurar.

Veri Modelleri

Veritabanı yaklaşımı, verinin düzenlenmesi ve depolanmasındaki ayrıntıların gizlenerek, verinin daha iyi anlaşılmasını sağlayan bir soyutlama düzeyi sunar. Bu soyutlamayı gerçekleştirmek için kullanılan araçlardan biri de veri modelidir. Veri modeli, bir veritabanının mantıksal yapısını tanımlayan kavramlar, işlemler ve kurallar bütünüdür. Bu yapının içeriği genellikle veri tipleri, ilişkiler ve kısıtlamalar gibi unsurları içerir. Veri modeli ayrıca verinin geri çağırılması ve güncellenmesi gibi işlemleri de içerebilir. Bu özellikler, veritabanı tasarımında esneklik ve anlaşılabilirlik sağlar.

Veri Modellerinin Sınıflandırılması

Yüksek düzeyli veya kavramsal veri modelleri, kullanıcıların veri algısını temsil eden kavramları içerir. Fiziksel veri modelleri ise bilgisayar ortamında verinin nasıl depolanacağına dair detayları tanımlar ve genellikle bilgisayar uzmanları için tasarlanır. Temsili modeller veya uygulama veri modelleri ise bu iki uç model arasında yer alır; hem son kullanıcılar hem de verinin depolanmasıyla ilgili kavramları içerir.

Kavramsal veri modelleri, varlık, öznitelik ve ilişkiler gibi kavramlardan yararlanır. Uygulama veri modelleri ise genellikle ticari VTYS'lerinde kullanılan modellerdir ve ilişkisel modeller gibi çeşitli yapıları içerir. Nesneye yönelik modeller, kavramsal veri modellerine yakın olan yüksek düzeyli uygulama modellerinin yeni bir türüdür ve ODGM standartlarına uyar. Nesneye yönelik veri modelleri aynı zamanda yazılım mühendisliğinde yüksek seviyeli kavramsal modeller olarak kullanılabilir.

Şemalar, Örnekler ve Veritabanının Durumu

Veritabanı şeması, veritabanının tasarımını tanımlayan bir yapıdır ve genellikle değişmez. Şemalar genellikle diyagramlarla gösterilir, ancak tüm detayları kapsamazlar. Veritabanındaki mevcut veriler sık sık değişebilir ve her anlık durumuna "anlık görüntü" denir. Her şema yapısı, kendi oluşum kümesine sahiptir. VTYS, şema yapılarını ve kısıtlarını tanımlar ve yazılım ihtiyaç duyduğunda bu şemalara başvurur. Bu şekilde, veritabanının yapılandırılması ve VTYS tarafından kullanılması sağlanır.

Üç Şema Mimarisi

Üç şema mimarisi, kullanıcı uygulamalarını fiziksel veritabanından ayırmayı amaçlar. İçsel düzey, veritabanının fiziksel depolama yapısını tanımlar. Kavramsal düzey, tüm veritabanının yapısını kullanıcı topluluğu için tanımlar ve fiziksel depolama ayrıntılarını gizler. Dışsal düzey, kullanıcıların ilgilendiği veritabanı bölümlerini tanımlar ve diğer kısımları gizler. Her düzey, farklı kullanıcı gruplarının gereksinimlerini karşılayan şemaları içerir. Bu sayede veritabanı yapısının karmaşıklığı gizlenir ve kullanıcılar için daha erişilebilir hale gelir.

Veri Bağımsızlığı

1. **Mantıksal veri bağımsızlığı** (logical data independence), kavramsal şemanın dışsal şemalarda veya uygulama programlarında değişiklik yapmadan değiştirilebilmesi, veritabanının genişletilmesi, kısıtların değiştirilmesi veya veritabanının daraltılması gibi işlemleri içerir. Bu, kavramsal şemanın esnekliğini ve veritabanı yapısındaki değişikliklerin kullanıcıların etkilenmemesini sağlar.
2. **Fiziksel veri bağımsızlığı** (physical data independence), kavramsal şemada yapılan değişikliklerin içsel şemada etkileri vardır, ancak içsel şemada yapılan değişiklikler dışsal şemalarda değişiklik gerektirmez. İçsel şemada yapılan değişiklikler genellikle fiziksel dosyaların yeniden düzenlenmesi gerektirir. Bu tür değişiklikler, veritabanının performansını artırmak veya güncellemek için yapılan değişiklikler olabilir.

Veritabanı Yönetim Sistemlerinde Kullanılan Diller

VTYS'ler, farklı kullanıcı gruplarına uygun dil ve arayüz sağlamak amacıyla tasarlanmıştır. Veritabanı tasarımının tamamlanmasının ardından, kavramsal ve fiziksel şemalar belirlenir ve bu iki düzey arasındaki eşleştirmeler yapılır. VTYS'lerde genellikle veri tanımlama dili kullanılarak şemalar tanımlanır ve bu tanımlar VTYS kataloğunda depolanır. Veri tanımlarının depolanması, tanımların kalıcılığını sağlar ve kullanıcıların bu tanımları kullanmasını sağlar. Veri tanımlarının yer aldığı yapıya "veri sözlüğü" de denir.

Kavramsal ve fiziksel düzeyler arasında açık bir ayrım olduğu VTYS'lerde, veri tanımlama dili kavramsal şemayı belirlemek için kullanılırken, depolama tanımlama dili fiziksel şemayı belirlemek için kullanılır. Görünüm tanımlama dili ise üç şemalı mimarinin tam anlamıyla kullanıldığı yapıların görünümelerini belirlemek ve bunları kavramsal şemadaki eşleştirmelerle belirlemek için kullanılır. Bugün kullanılan VTYS'lerde, SQL gibi geniş kapsamlı diller genellikle tüm bu işlevleri yerine getiren tek bir dil olarak kullanılır.

Veritabanı Yönetim Sistemlerinin Bileşen Modülleri

Veritabanı ve VTYS kataloğu diske depolanır. Disk erişimi işletim sistemi tarafından kontrol edilir. VTYS, kendi ara bellek yönetim modülü aracılığıyla disk okuma/yazma işlemlerini programlar ve performansı artırır. Depolanan veri yöneticisi, VTYS'nin yüksek düzey modülüdür ve depolanan bilgilere erişimi yönetir.

Tipik VTYS bileşenlerini açıklayan Şekil 1.3, kullanıcı etkileşimlerini ve veri depolama işlemlerini iki bölümde gösterir. Üst kısımda, farklı kullanıcılar ve etkileşimleri yer alırken, alt kısımda veri depolama ve işlemeye ilgili bileşenler bulunur.

Üst kısımdaki kullanıcılar arasında yöneticiler, standart kullanıcılar ve uygulama programcıları yer alır. Yöneticiler, veri tanımlama dili ve özel komutlarla veritabanını yönetir. Standart kullanıcılar, etkileşimli sorgularla veritabanıyla iletişim kurarlar. Alt kısımdaki bileşenler, verilerin depolanması ve işlenmesinden sorumludur. Bu bileşenler, VTYS'nin fiziksel yapısını oluşturur ve verilerin yönetimini sağlar.

VERİTABANI TÜRLERİ

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

Hiyerarşik Veritabanı

Hiyerarşik veritabanları, eski ve yaygın bir veri modelidir. Veri yapısı ağaç benzeri bir yapıdadır, kök ve dallardan oluşur. Veriler arasındaki ilişkiler ebeveyn-çocuk ilişkisine benzer, bir ebeveyn birden fazla çocuğa sahip olabilirken, her çocuk yalnızca bir ebeveynle ilişkilidir. Bu modelde, veriye erişim kök düğümden başlar ve aşağı doğru devam eder. Silme işlemi bir dalın tüm alt dallarını ve bu dallara ait verileri de siler. Bu yapı, IMS gibi yazılımlar tarafından kullanılır ancak veri tekrarı ve erişim sorunları gibi zorluklar yaşanabilir.

Ağ Veritabanı

Ağ veritabanı, 1970'lerin sonu ve 1980'lerin başında kullanılan bir veritabanı türüdür ve hiyerarşik veri modelinin ebeveyn-çocuk ilişkisindeki kısıtlamaları gidermeyi amaçlar. Ağ veri modelinde her düğüm, birden fazla ebeveyn ve birden fazla çocuk düğümüne bağlanabilir, bu da hiyerarşik yapıdan farklıdır.

İlişkisel Veritabanı

İlişkisel veritabanı modeli, 1970'lerde ortaya atılmış ve 1970'lerin sonlarından itibaren kullanılmaya başlanmıştır. 1985'ten sonra ise kullanımı daha da yaygınlaşmıştır. Bu model, birden çok ilişki biçimini kullanabilme özelliğiyle diğer veri modellerinden farklılık gösterir. Günümüzde hemen hemen tüm veritabanı yönetim sistemlerinde tercih edilen bir modeldir.

Nesneye Yönelik Veritabanı

Günümüzde ve gelecekte kullanılan pek çok uygulama, yalnızca metin tabanlı verileri değil, aynı zamanda multimedya öğelerini de içeren veritabanı yönetim sistemlerine ihtiyaç duyar. Geleneksel veritabanı sistemleri, metin verilerini satır ve sütunlar şeklinde düzenlemek için tasarlanmıştır ve grafik veya multimedya unsurlarını yönetmek için uygun değildir. Bu eksikliği gidermek için nesneye yönelik veri modelleri ve nesneye yönelik veritabanları geliştirilmiştir. Bu veritabanları, metin ve multimedya öğelerini daha etkin bir şekilde işlemek için tasarlanmıştır.

VERİTABANI YÖNETİM SİSTEMİ YAZILIMLARI

Günümüzde yaygın olarak kullanılan veritabanı yönetim sistemleri şunlardır:

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

Oracle: Oracle firması tarafından geliştirilmiş bir ilişkisel veritabanı yönetim sistemi yazılımıdır. Kurumsal amaçlar için tercih edilen bu yazılım, büyük veri miktarlarını çok kullanıcı ortamlarında saklama ve güvenli erişim sağlama yetenekleriyle bilinir. Birçok işletim sistemi üzerinde kullanılabilir, ancak maliyeti yüksektir.

MySQL: Açık kaynak kodlu bir veritabanı yönetim sistemi yazılımıdır. Unix, OS/2 ve Windows gibi işletim sistemlerinde ücretsiz olarak kullanılabilir. Özellikle web ortamlarında yaygın olarak kullanılır ve kullanımı kolay bir kullanıcı yetkilendirme sistemine sahiptir. Python'dan Java'ya kadar birçok programlama diliyle erişilebilir.

Sybase: SAP'a baęlı olan Sybase firması tarafından geliştirilmiş bir veritabanı yönetim sistemi yazılımıdır. Orta ve büyük ölçekli uygulamalar için tercih edilir ve özellikle bankacılık sektöründe ve kamusal alanlarda kullanılır. Veritabanı işlemleri genellikle T-SQL sorgu dili ile gerçekleştirilir.

PostgreSQL: PostgreSQL, açık kaynak kodlu ve ücretsiz bir yazılımdır. Yeni nesil veritabanı yönetim sistemi araştırma prototipi olan POSTGRES'in geliştirilmiş halidir. Zengin veri tipleri ve güçlü veri modeliyle beraber SQL'in geliştirilmiş alt kümesi olan PostQuel dilini kullanır. UNIX ve UNIX türevi işletim sistemlerinde çalışırken, aynı zamanda Microsoft Windows NT tabanlı işletim sistemlerinde de kullanılabilir.

MS Access: Microsoft Office paketinin bir parçası olan MS Access, küçük ölçekli uygulamalarda masaüstü veritabanı sistemi olarak kullanılır. Yalnızca Windows işletim sisteminde çalışır ve işletim sisteminin sağladığı güvenlik seçeneklerini kullanır.

DB2: IBM tarafından geliştirilen DB2, bir veritabanı yönetim sistemi yazılımıdır. Küçük ve büyük ölçekli işletmeler için tercih edilir ve Windows, Linux ve Unix işletim sistemlerinde kullanılabilir.