

Bilgisayar Programcılığı 2.Sınıf I.Öğretim

No: 223010710101

Ad Soyad:Beşşar Elhasan

Veritabanının Tarihçesi

Bir konuyla ilgili verilere hızlı ve kolay biçimde erişebilme isteği, verilerin düzenli biçimde saklanması ihtiyacını doğurmuş, dolayısıyla veritabanı olgusunun bu adla olmasa bile geçmişten günümüze kadar bir biçimde hayatımızda yer almasına neden olmuştur.

Kütüphanelerde herhangi bir kitaba erişmek için kullanılan kitap künye bilgisi bu tür düzenlemelere örnek olarak verilebilir.

Düzenlenmesi ve saklanması gereken veri miktarının giderek artması ve bu verilerin fiziksel olarak uzun süre saklanabilmesi, korunabilmesi ve istendiğinde hızlıca erişilebilmesi kolay bir iş değildi.

Bilgisayar kullanımının özel işletmeler için uygun maliyetli bir seçenek olmaya başladığı dönem olan 1960'larda veritabanlarının bilgisayarlar yardımıyla oluşturulması dönemi de başlamıştır.

1969'da Edgar F. Codd tarafından geliştirilen ilişkisel veritabanı modeli, insanların veritabanları hakkındaki düşüncelerini değiştirecek nitelikte bir gelişme olarak ortaya çıkmıştır.

Modelde, veritabanının şeması veya mantıksal organizasyonu, fiziksel bilgi deposundan bağımsız hâle getirilmiş ve bu da veritabanları için standart bir prensip hâline gelmiştir.

Buna paralel olarak ilişkisel veritabanı sistemleri, bilgisayar satışlarındaki hızlı yükselişin veritabanı piyasasını canlandırmasıyla ticari bir başarıya dönüşmüştür.

Veritabanına ilişkin temel kavramlar

Günümüzde sıklıkla karşılaşılan bir kavram hâline gelen veritabanı çok genel anlamda, bir kurumun ihtiyaç duyduğu ve kullandığı veriler bütünüdür.

Birbirleriyle ilişkilendirilip yorumlanmadıkları sürece tek başlarına bir anlam ifade etmezler ve bu hâllerleriyle karar verme konusunda da karar vericilere bir katkı sağlayamazlar.

Veritabanı, herhangi bir konuda birbiriyle ilişkili olan ve amaca uygun olarak düzenlenmiş, mantıksal ve fiziksel olarak tanımlanmış veriler bütünüdür.

Veritabanı herhangi bir kurumda birden fazla uygulamada ortak olarak kullanılabilen verilerden oluşur.

Bir diğer örnek olarak öğrenci bilgilerinin yer aldığı bir dosyada, dosyanın en sonunda yer alan bir öğrenci bilgisine erişmek için dosyanın tamamının okunması gerekecektir.

Bu erişim biçiminde, verilerin yer aldığı fiziksel adresler birer indeks numarası ile tanımlanıp bu indeks numaraları da ayrı bir dosya olarak saklanır.

Herhangi bir veriye erişilmek istendiğinde, önce indeks dosyasından verinin yer aldığı adresi gösteren indeks numarası bulunur, daha sonra bu numaraya karşılık gelen fiziksel adrese doğrudan erişim sağlanır.

GELENEKSEL DOSYA SİSTEMLERİ VE VERİTABANI YÖNETİM SİSTEMLERİ

Verilerin belirli bir düzen içinde saklanması ve ihtiyaç duyulduğunda erişilebilmesi amacıyla bilgisayar kullanımında geçmişten günümüze farklı yaklaşımlar benimsenmiştir.

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. Bununla birlikte iki farklı dosya içinde aynı verinin yer alması da söz konusu idi.

F verisinin ise yalnızca üretim bölümünün ihtiyaç duyduğu bir veri olduğu ve diğer bölümlerin veri dosyalarında yer almadığı farkedilecektir.

Aynı verinin birçok dosya içinde tekrarlanması veri yönetimiyle ilgili pek çok zorluğu da beraberinde getirecektir.

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.

Geleneksel dosya sistemlerinin sakıncaları Veritabanı yönetim sistemlerinin üstünlükleri Veri tekrarı ve veri tutarsızlığına yol açar.

Veri tekrarı ve veri tutarsızlığını önler.

VTYS için gerekli olan donanım, yazılım ve veritabanı eğitimi için başlangıç yatırımlarının yüksek olması, veri güvenliğini sağlama, aynı anda gerçekleştirilen işlemlerin kontrolü, veri kurtarma ve bütünlük fonksiyonları için gerekli olan sabit maliyetlerin yüksek olması VTYS'nin dezavantajlı yönlerini oluşturmaktadır.

VERİTABANI YÖNETİM SİSTEMLERİNİN AVANTAJLARI

Tablo 1.1’de verilen özelliklere ilişkin ayrıntılı açıklamalar, veritabanı yönetim sistemlerinin üstünlüklerine vurgu yapacak biçimde izleyen kesimde verilmiştir.

Gereksiz Veri Tekrarı ve Veri Tutarsızlığının Önlenmesi

Böylece verilerden bir kısmı bilgisayarda aynı anda birden fazla dosya içinde yer alabilmekte dolayısıyla verinin büyük ölçüde tekrar edilmesi sonucuyla karşılaşılmaktadır.

Bu durum aynı zamanda veri derleme işleminin de tekrarlı olmasına ve veri derleme için yapılan harcamaların artmasına neden olabilmektedir.

Diğer yandan veri tekrarı, verilerin depolanması için kullanılan bellek kapasitelerinin dolayısıyla donanım harcamalarının artmasına da yol açar.

Veri tekrarı aynı zamanda tekrarlanan verilerin farklı dosyalardaki değerlerinin de farklı olabilmesine ve dolayısıyla veri tutarsızlığı ile uygulamalarda sorunlar yaşanmasına neden olabilir.

Örneğin, bir işletmede yer alan ürünlere ilişkin stok numarası bazı dosyalarda stok kodu olarak nitelendirilmiş olabilir. Aynı işletmenin stoklar, satışlar ve giyim mağazalarına ilişkin veri dosyalarında aynı ürüne ilişkin ölçü birimi olarak bir dosyada XL kodunun bir başka dosyada 3 rakamı ile belirtilmesi tutarsızlığa bir başka örnektir.

Veritabanı yönetim sistemi kullanımıyla; veriler arası ilişkileri kurmak için T.C.Kimlik numarası, personel sicil numarası ve stok kodu gibi ayırtedici nitelikte olan veri değerlerinin zorunlu tekrarlanması dışında, veri tekrarı önlenir.

Veri Bütünlüğünün Sağlanması

Veri bütünlüğü, veritabanında yer alan bir verinin farklı uygulamalarda kullanımı sözkonusu olduğunda veri üzerinde yapılacak bir değişimin verinin kullanıldığı diğer uygulamalara da yansıtılması anlamına gelir.

Bu, özellikle verinin silinmesi durumunda önem kazanmaktadır.

Buna göre bir veri silindiğinde verinin ilişkili olduğu tüm uygulamalardan o verinin silinmesi gereklidir.

Geleneksel dosya sistemlerinde veri bütünlüğünün sağlanması güç iken veritabanı yönetim sistemlerinde, sisteme girilen kısıtlamalar ve kurallar yardımıyla veri bütünlüğü rahatlıkla sağlanabilmektedir.

Veri Paylaşımının Sağlanması

Geleneksel dosya sistemlerinde aynı veriye birden fazla kullanıcının aynı anda erişmesine olanak yoktur.

Buna göre veriye erişme isteğini ilk bildiren ilk sırada veriye erişir, diğer kullanıcının veriye erişebilmesi için ilk kullanıcının işinin bitmesi gerekir.

Veritabanı yönetim sistemlerinde ise sunucu/istemci mimarisi kullanılarak veriler tek bir merkezden, ağ üzerinde yer alan ve erişim yetkisi olan tüm kullanıcılara aynı anda veriye erişme olanağı sunar.

Bununla ilgili olarak bankacılık sektöründen bir örnek verebiliriz.

Bir bankanın veritabanı tanımları arasında "Bir banka hesabından, hesaptaki mevcut miktardan daha fazla para çekilemez, hesap bakiyesi negatif olamaz" biçiminde ifade edilebilecek bir kısıtlamanın bulunduğunu varsayalım. Bu bankadaki bir hesapta 200.000 TL bulunduğunu ve bu hesaptan para çekmeye yetkili iki kişinin bulunduğunu düşünelim.

Bu olumsuz durum veritabanı üzerinde tanımlanan bir kısıtlamayla önlenir. Kullanımda Üst Düzey Uzmanlık Gerektirmemesi

Günümüzde kurumlar, bilgi sistemlerinden yoğun bir biçimde yararlanmakta ve bu bilgi sistemleri kapsamında birbiriyle ilişkili çok çeşitli ve büyük miktarda veriler üzerinde her türlü güncelleme, sorgulama, raporlama vb. Verilerin karmaşıklığı, veriler arasında çeşitli düzeylerde birçok ilişkilendirmenin yapılması gerekliliği ve uygulamaların çeşitliliği nedeniyle, kullanılan dosya yapıları, dosyalar arası ilişkileri kurmak için kullanılan teknikler ve dosyalara erişim için kullanılan yaklaşımlar oldukça karmaşık bir yapı sergilemektedir.

Bu nedenle sözü edilen uygulama programlarının geliştirilmesi için bilişim teknolojisinde uzman kişilere ihtiyaç duyulmaktadır.

Çeşitli verilere ulaşma ihtiyacında olan çok sayıda kullanıcı kesiminin bilgi sistemi ile etkileşimi de yalnızca sözkonusu bu uygulama programları ile sağlanabilmektedir.

Özellikle yönetici konumundaki kullanıcılar önceden belirlenmemiş çok seyrek ya da yalnız bir kez oluşan bilgi sistemi ihtiyaçlarını bu biçimde tanımlanmış uygulama programlarıyla karşılamakta zorlanırlar.

Bunlardan birincisi, VTSY'lerin kullanıcıları verilerin saklanması için bilgisayar belleklerinde oluşturulan karmaşık fiziksel yapılarla ilgilenmek zorunda bırakmayıp, yalnızca ilgilendikleri verileri içeren basit mantıksal yapıları görmesine olanak sağlamasıdır.

Bunun sonucunda önceden tanımlanmamış veri ihtiyaçlarını karşılamak için bilişim uzmanına olan gereksinim ortadan kalkmış ve ilgili kullanıcılar, sorgu dillerini kullanmak yoluyla veri ihtiyaçlarını karşılayabilir duruma gelmiştir.

Verilerin Gizliliğinin ve Güvenliğinin Sağlanması

Herhangi bir kurumda, kurum çalışanları tarafından ortak kullanılan verilerin depolandığı yapının ve bu yapı içinde yer alan verilerin gizliliği ve güvenliği çok önemli bir konudur.

Ancak her veri üzerinde her kullanıcının bu faaliyetlerin tamamını gerçekleştirebilmesi, istenen bir durum değildir. Örneğin bir okulda öğrencilere ilişkin verilerle ilgilenen farklı kullanıcı grupları bulunur.

Bu kullanıcılardan bir kısmı veriler üzerinde yalnızca okuma işlemi gerçekleştirebilirken, bir kısmı hem okuma hem de yazma, diğer bir kısmı ise okuma, yazma ve güncelleme işlemlerini yapabilir.

Sisteme yanlış girilen bir notun düzeltilmesi ise bir dizi dilekçe ve alınan kararlar sonucu öğrenci işleri daire başkanlığındaki ilgili personel tarafından gerçekleştirilebilir.

Herhangi bir kullanıcı veritabanı üzerinde işlem yapmak istediğinde, veritabanı öncelikle kullanıcının yapmak istediği işleme ilişkin yetkisinin olup olmadığını kontrol eder.

Kullanıcı yetkisiz olduğu bir işlem yapmak istiyorsa veritabanı yönetim sistemi bu işlemin yapılmasına izin vermez.

Standart Yapı ve Kuralların Uygulanabilir Olması

Geleneksel dosya sistemlerinde her alt sistem kendi faaliyetlerine özgü uygulamaları ve her uygulama da kendi dosya yapısını kullandığı için dosya yapılanmalarında belirli bir standart olmayıp farklılıklar bulunmaktadır.

VTSY'lerde ise merkezi bir kontrol sistemi bulunur.

VTSY'lerin yapısal özellikleri ve veritabanı sorumlusunun varlığı veriler ve veritabanı üzerinde belirli standartların oluşturulması ve uygulanabilmesini olanaklı kılar.

Bu standartlar verinin yapısı, gösterim biçimi, adlandırılması, belgelenmesi ile ilgili yapısal standartlar olabileceği gibi kurum içi, kurumlar arası, ulusal ya da uluslararası düzeyde belirlenmiş kurallar biçiminde de olabilir.

Standartların varlığı, veritabanı yapısını anlama ve kullanma açılarından büyük kolaylık sağlamanın yanısıra farklı sistemler arasında veri alışverişi için de çok önemli bir ihtiyaçtır.

VERİTABANI KULLANICILARI

Veritabanı ile herhangi bir şekilde etkileşimde olan kişi ya da kişiler veritabanı kullanıcısı olup aşağıdaki gibi 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

Veritabanı sorumluları, veritabanının tasarlanması, oluşturulması ve veritabanının işletim faaliyetlerinden birinci derecede sorumlu olan ve veritabanı üzerinde en fazla yetkiye sahip olan kullanıcılarıdır.

Veritabanı sorumluları, veritabanı yöneticisi ve veritabanı tasarımcısı olarak iki başlık altında incelenebilir.

Her iki sorumluluğu aynı kişi ya da kişiler alabileceği gibi veritabanını kullanacak olan kurumun/veritabanının büyüklüğü ve kullanıcı sayısı gibi faktörlere bağlı olarak veritabanı yöneticisi ile veritabanı tasarımcısının farklı kişiler olması da mümkündür.

Veritabanı Yöneticisi

Veritabanı yöneticisinin 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.

Büyük işletmelerde bu sorumluluklar için yardımcı personele de ihtiyaç duyulur.

Veritabanı Tasarımcısı

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

Veritabanı tasarımcısı tasarım boyunca potansiyel kullanıcı gruplarıyla karşılıklı etkileşim hâlinde olup bu grupların veriye erişimini ve veri üzerinde işlem yapabilmelerini olanaklı kılan kullanıcı görünümlerini geliştirirler.

Bütünlük kısıtlamalarını belirleyip tanımlama: Veritabanında veri bütünlüğünün sağlanabilmesi, veri kaybının önüne geçilebilmesi, veri bütünlüğünü tehlikeye sokacak kullanıcı hatalarının önlenmesi amacıyla gerekli kurallar, ilişkiler ve kısıtlamaların belirlenmesi.

Veritabanı kullanım yetkilerini tanımlama: Veritabanı kullanıcılarının ve kullanım yetkilerinin tanımlanması ile her kullanıcı grubunun hangi veriler üzerinde hangi işlemleri yapmaya yetkili olduğunun belirlenmesi.

Veritabanı güvenliğini sağlama: Kullanıcıdan ya da yazılım sorunlarından kaynaklanabilecek veri kaybının önlenmesi amacıyla veritabanının yedeklerinin alınması ve kurtarma işlemlerine ilişkin düzenlemelerin yapılması.

Veritabanının işletimini izleme ve sürekliliğini sağlama: Veritabanının kullanıma sürekli açık olmasının sağlanması ve sistemde meydana gelebilecek herhangi bir sorunu hızlı biçimde giderebilecek tedbirlerin alınması.

Veritabanından beklenen performans sağlama: Veritabanı kullanıcılarının beklentilerine cevap verecek bir yapının donanım ve yazılım ihtiyaçlarının sağlanması, veritabanındaki tanımlamaların, kısıtlamaların kullanıcı ihtiyaçlarına cevap verebilecek yeterlilikte olup olmadığının izlenmesi, gerekiyorsa değişikliklerin yapılması.

SON KULLANICILAR

Son kullanıcılar da kendi içinde gruplandırılabilir: Standart son kullanıcılar: Veritabanına nadiren erişim yapan fakat her seferinde farklı bilgi ihtiyacı olabilen kullanıcılarıdır.

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 öğeler yardımıyla kullanım kolaylığı sağlayan hazır paket programlarını kullanarak kişisel veritabanlarının sürekliliğini sağlar.

Standart son kullanıcılar da VTYS özellikleri içinden yaptıkları işin gereği olarak, yalnızca sürekli tekrarlayacakları özellikleri öğrenirler.

Gelişmiş son kullanıcılar ise karmaşık gereksinimlerini karşılayabilmek için VTYS'nin pek çok özelliğini öğrenmeye çalışırlar.

Bağımsız son kullanıcılar da veritabanı ihtiyaçlarını karşılamak üzere belirli bir yazılım paketini kullanma konusunda ustalaşırlar.

Sistem Analistleri ve Uygulama Programcıları

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, hataları ayıklayan, belgeleyen ve kaydedilmiş işlemler olarak sürekliliğini sağlayan kişilerdir.

Yaygın olarak yazılım geliştiriciler ya da yazılım mühendisleri olarak da anılan analistler ve programcıların yukarıda sıralanan görevlerini yerine getirebilmeleri için VTYS'nin sağladığı tüm özellikleri bilmeleri gerekir.

VERİTABANI YÖNETİM SİSTEMLERİNİN MİMARİSİ

VTYS'lerinin mimarisi geçmişten günümüze incelendiğinde, ilk veritabanı sistemlerinde, VTYS'nin tüm yazılım paketlerinin tek bir sisteme entegre edildiği, modern VTYS'lerinde ise istemci/sunucu mimarisi ile modüler bir yapılanmanın sözkonusu olduğu görülmektedir.

Buna bağlı olarak, büyük merkezi ana bilgisayarlar yerini, iletişim ağları aracılığıyla çeşitli sunucu bilgisayarlara bağlanan yüzlerce iş istasyonu ve kişisel bilgisayarlara bırakmıştır.

Burada sözü edilen sunucu bilgisayarlar; web sunucular, veritabanı sunucuları, dosya sunucuları, uygulama sunucuları vb.

Basit bir istemci/sunucu VTYS mimarisinde sistem fonksiyonel olarak iki modüle ayrılmaktadır.

İstemci modülü, VTYS'nin herhangi bir kullanıcı iş istasyonunda ya da kişisel bilgisayar üzerinde çalışan parçasıdır.

Tipik olarak istemci modülünde veritabanına erişmek için uygulama programları ve kullanıcı arayüzleri bulunur.

Sunucu modülünde ise veri deposu ile veri deposuna erişimi ve sorgulamayı sağlayacak fonksiyonlar yer alır.

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.

Veri modellerinin çoğu verinin geri çağırılması ve veritabanı üzerinde güncelleme yapmaya yönelik işlemleri de içerir.

Veri modelinin bu temel işlevlerine ek olarak veritabanının dinamik hareketlerini belirleyen kavramları içermesi de yaygınlaşmıştır.

Veri Modellerinin Sınıflandırılması

Yüksek düzeyli ya da kavramsal veri modelleri (conceptual data models), kullanıcıların veri algılama biçimiyle ilişkili kavramları kapsar.

Düşük düzeyli ya da fiziksel veri modelleri (physical data models), 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 (implementation data model) 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. Bu kapsamda varlık-ilişki modelleme konusu 2.Ünite’de ele alınmıştır. Uygulama veri modelleri geleneksel ticari VTYS’lerinde en sık kullanılan modellerdir. Bu modeller geçmişte

yaygın olarak kullanılan ağ ve hiyerarşik veri modeli ile günümüzde yaygın olarak kullanılan ilişkisel modelleri içerir. Uygulama veri modelleri, verileri kayıt yapılarıyla gösterdiklerinden bazen kayıt tabanlı veri modelleri olarak da adlandırılırlar. Nesneye yönelik modeller kavramsal veri modeliyle yakın ilişkili olan yüksek düzeyli uygulama modellerinin yeni üyesi olarak düşünülebilirler. Nesneye yönelik veri modelleri için bir standart ODGM (Object Data Management Group-Nesneye Yönelik Veri Yönetim Grubu) olarak adlandırılır. Nesneye yönelik veri modelleri aynı zamanda (özellikle yazılım mühendisliği alanında) yüksek seviyeli kavramsal model olarak da kullanılmaktadır.

Şemalar, Örnekler ve Veritabanının Durumu

Veritabanının herhangi bir veri modeliyle tanımlanması veritabanı şeması olarak adlandırılır.

Veri modellerinin çoğunda şemaların bir diyagram olarak gösterilebilmesi amacıyla kullanılan belirli kuralları vardır.

Örneğin, her bir veri kalemine ilişkin veri tipi, çeşitli dosyalar arasındaki ilişkiler ya da verilere ilişkin kısıtlar diyagramından izlenemez.

Örneğin, bir öğrenci veritabanında her yeni öğrenci kaydında veritabanı değişecektir.

Bir veritabanının herhangi bir andaki durumuna anlık görüntü denir.

Veritabanı durumu, belirli bir şema ile ilişkilendirilerek yapılandırılır.

Her zaman yeni bir kayıt ekleme, silme ya da bir kayıttaki verinin değerini değiştirmek mümkündür.

Üç Şema Mimarisi

Kavramsal düzey, kullanıcı topluluğu için tüm veritabanının yapısını tanımlayan kavramsal şemayı içerir.

Bu kavramsal şema, yüksek düzeyli kavramsal veri modelleri kullanılarak tasarlanır. Dışsal düzey, bir dizi dışsal şema ya da kullanıcı görünümü içerir.

Her dışsal şema, bir yüksek düzey veri modelinde tasarlanan dışsal şema tabanlı uygulama veri modeli kullanır.

Örneğin bazı eski VTYS'leri fiziksel düzey ayrıntılarını ayrı bir düzey olarak ele almayıp kavramsal şema içinde barındırmaktaydı.

Bu nedenle VTYS, dış şemada yapılan belirli bir isteği kavramsal şema isteğine daha sonra da depolanmış veritabanı üzerinde işlem yapmak üzere içsel şema isteğine dönüştürmek zorundadır.

Ema düzeyleri arasında bu şekilde gerçekleştirilen dönüşümler eşleştirme olarak adlandırılır.^{[1][2]}

Veri Bağımsızlığı

Ema düzeyleri arasındaki dönüştürme kapasitesi olarak tanımlanabilen veri bağımsızlığı yine şema mimarisi ile açıklanabilir.

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

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. Fiziksel veri bağımsızlığı, fiziksel ayrıntıların olduğu çoğu veritabanı ve dosya ortamında mevcuttur. Bununla birlikte kavramsal veri bağımsızlığına erişmek daha zordur çünkü kavramsal veri bağımsızlığı uygulama programlarını etkilemeden yapısal değişiklikler ve kısıtlama değişikliklerine izin verir. Çok düzeyli VTSY'leri söz konusu olduğunda bu VTSY'lerin, farklı düzeyler arasındaki veri ve istek eşleştirmelerini haritalayan bir kataloğu da barındırması gerekir.

Veritabanı Yönetim Sistemlerinde Kullanılan Diller

Düzyerler arasında mutlak bir ayrımın yapılmadığı pekçok VTYS'de iki şemayı tanımlamak amacıyla veri tanımlama dili olarak adlandırılan bir dil, veritabanı yöneticisi ve veritabanı tasarımcısı tarafından kullanılır. Veritabanında veri tanımlarının yer aldığı yapı veri sözlüğü olarak da adlandırılabilir. Veritabanı şemaları yukarıda sözü edilen biçimde tanımlanıp derlendiğinde ve ilgili veriler veritabanına kaydedildiğinde veritabanı, üzerinde veri düzenleme işlemi yapılabilir duruma gelir. Burada sözü edilen veri işleme; veritabanından veri çağırma, veri ekleme, veri silme ve veri üzerinde değişiklikler yapma faaliyetlerini içerir.

Veritabanı Yönetim Sistemlerinin Bileşen Modülleri

Bu kısım, veritabanı yönetici personelleri, sorguları formüle etmek için etkileşimli ara yüzlerle çalışan standart kullanıcılar, programlama dillerini kullanarak program oluşturan uygulama programcıları ve önceden tanımlanmış işlemler için parametreler sağlayarak veri giriş işlerini yapan sıradan kullanıcılar için ara yüzleri göstermektedir. Veri tanımlama dili derleyicisi, veri tanımlama dilinde belirlenmiş şema tanımlarını işler ve şemaların tanımlarını VTYS kataloğuna depolar.

Katalog; dosyaların isimleri ve boyutları, veri kalemalarının adları ve veri türleri, her dosyanın depolama ayrıntıları, şemalar arasındaki bilgi eşleştirmesi ve kısıtlamalar gibi bilgileri içerir. Depolanmış veri hakkında istatistiki ve diğer fiziksel bilgiler için sistem kataloğuna başvurarak sorgu için gerekli işlemleri uygulayan yürütülebilir kodları üretir.

VERİTABANI TÜRLERİ

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

Bu yapı, başlangıç noktası ağacın kökü, bağlantılı kılınacak noktalar dallar ve ana dallara bağlı alt dallar olarak düşünülebilen bir yapı biçimindedir.

Buna göre veriler arasındaki ilişkilerde hiyerarşinin üst bölümünde olan dallar alt bölümde birden çok dal ile bağlantılı olabilirken alt bölümünde olan dallar üstte kalan dallara yalnızca tek bir noktadan bağlantılı olabilirler.

Ağ Veritabanı

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 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

Bu nedenle bu eksikliği gidermek amacıyla nesneye yönelik veri modelleri geliştirilmiştir.

Sözkonusu bu modelleri kullanan veritabanları da nesneye yönelik veritabanı olarak adlandırılmaktadır.

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

MS SQL Server: Microsoft firması tarafından geliştirilmiş ve ilişkisel veri modelini kullanan veritabanı yönetim sistemi yazılımıdır.

MySQL: Açık kaynak kodlu bir veritabanı yönetim sistemi yazılımıdır.

Sybase: Avrupanın en büyük yazılım şirketi olan SAP'a bağlı Sybase firması tarafından geliştirilen veritabanı yönetim sistemi yazılımıdır.

Yi performans veren, güvenli ve geniş özelliklere sahip bir veritabanı yönetim sistemi yazılımıdır.

MS Access: Microsoft firmasının Microsoft Office yazılım paketi içinde yer alan veritabanı yönetim sistemi yazılımıdır.