

HİTİT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

Veri Tabanı Yönetim
Sistemleri

İÇİNDEKİLER

1. Veri Tabanı Temel Kavramları <i>Doç. Dr. ABDULKAD R ÖZDEM R</i>	4
2. Veri Tabanı ve Normalizasyon <i>Doç. Dr. ABDULKAD R ÖZDEM R</i>	24
3. Veri Tabanı Araçlarının Kurulumunu Yapmak <i>Dr. Ö r. Üyesi SERDAR AYDIN</i>	43
4. Tabloları Olu turmak ve Özelliklerini Belirlemek <i>Dr. Ö r. Üyesi SERDAR AYDIN</i>	67
5. TSQL ile Veri Tabanı ve Tabloları Olu turmak, Özelliklerini Belirlemek <i>Dr. Ö r. Üyesi SERDAR AYDIN</i>	91
6. Sorgu Olu turmak ve Çe itlerini Kullanmak-1 <i>Dr. Ö r. Üyesi SNAN KUL</i>	112
7. Sorgu Olu turmak ve Çe itlerini Kullanmak-2 <i>Dr. Ö r. Üyesi SNAN KUL</i>	136
8. Sorgu Olu turmak ve Çe itlerini Kullanmak - 3 <i>Dr. Ö r. Üyesi SNAN KUL</i>	157
9. Sorgu Olu turmak ve Çe itlerini Kullanmak - 4 <i>Ar. Gör. YAKUP BAYO LU</i>	180
10. İli kili Tablolar ile Sorgu Hazırlamak <i>Ar. Gör. YAKUP BAYO LU</i>	203
11. Görüntü (VIEW), Store Prosedür ve Fonksiyonlar <i>Ar. Gör. YAKUP BAYO LU</i>	223
12. Veri Tabanı Yönetimi Yapmak <i>Dr. Ö r. Üyesi AHMET KAM L KABAKU</i>	248
13. Veri Tabanı Güvenli ini Sa lamak <i>Dr. Ö r. Üyesi AHMET KAM L KABAKU</i>	271
14. Veri Tabanında Yedekleme ve Geri Yükleme <i>Dr. Ö r. Üyesi AHMET KAM L KABAKU</i>	291

VERİ TABANI TEMEL KAVRAMLARI



- HEDEFLER
 - Veri Tabanı ile Dosyaların Karşılaştırılması
 - Veri Tabanı Tanımı
 - Veri Modelleri
 - Veri Tabanı Veri Türleri
 - Veri Tabanı Tabloları
 - Veri Tabanı Kullanıcıları

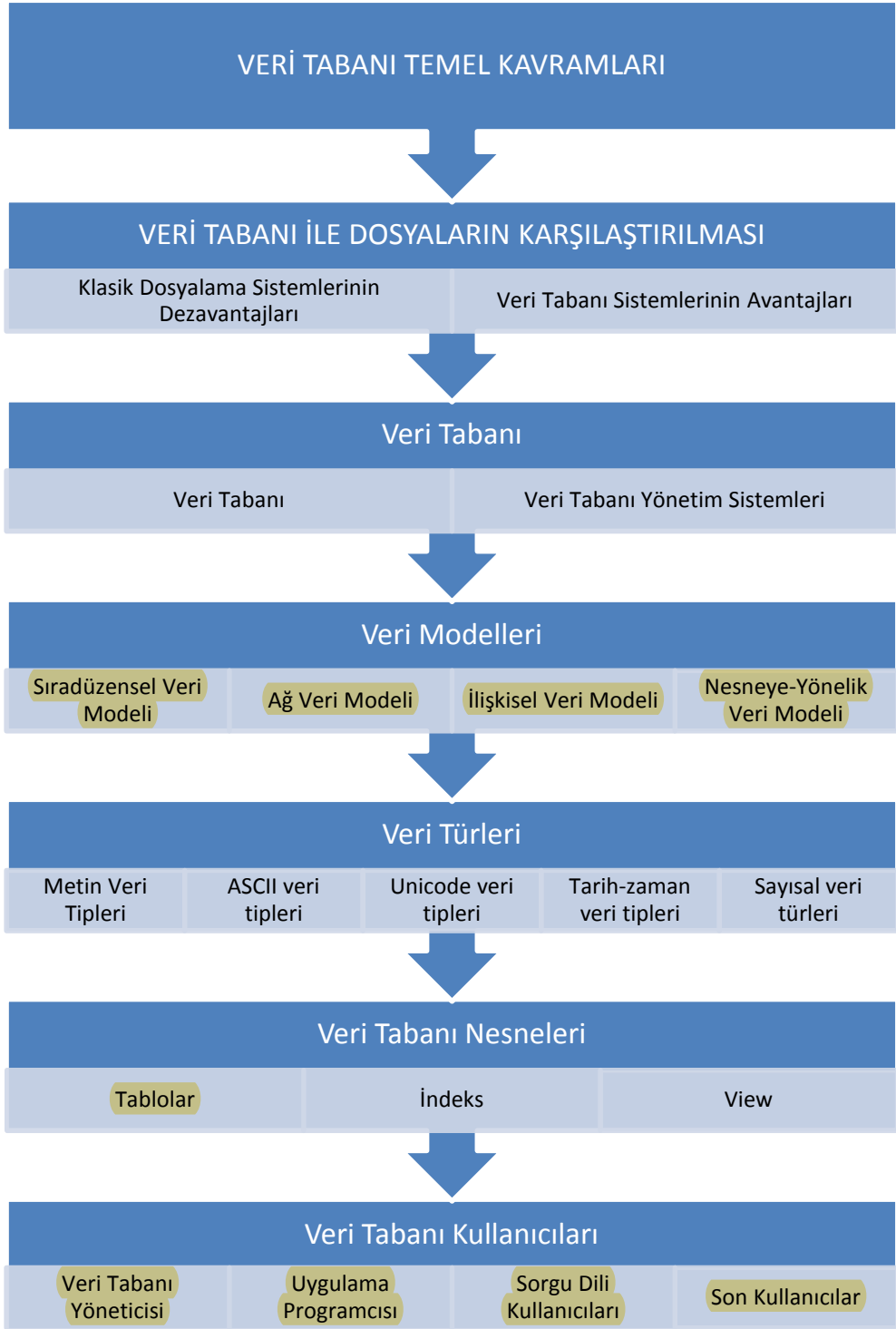
VERİ TABANI YÖNETİM SİSTEMLERİ

İÇİNDEKİLER

HEDEFLER

- Bu üniteyi çalıştıktan sonra;
- Neden Veri Tabanı kullanmanız gerektiği hakkında fikir sahibi olabileceksiniz,
- Veri Tabanı türleri ile veri tabanlarında kullanılan veri türlerini kavrayabileceksiniz,
- Veri Tabanında kullanılan tablolar hakkında bilgi sahibi olabileceksiniz,
- Veri Tabanı kullanıcılarının yaptığı görevlerin farkını anlayabileceksiniz.

ÜNİTE 1



GİRİŞ

Bilgisayar teknolojilerindeki en büyük gelişmelerden biri de saklama ortamlarındaki kapasite artışıdır. Bu kapasite artışları aslında verilerdeki artış ve bu artan verilerin saklanması ihtiyacından kaynaklanmıştır. 1980'lerde 100 MB seviyelerinde bulunan sunucu Hard Disk kapasiteleri günümüzde 100 TB mertebelerini geçmiştir. Bu artış bir milyon kat artışı ifade etmektedir. Bu kapasite artışı verilerdeki artışın da başka bir göstergesidir.

Veri artışı aslında verilerin saklandığı klasik dosyalama yapısının sınırlarını zorlamakta ve bazen de verilerin yönetilmez hâlini almasını sağlamaktaydı. Bu zorlukların üstesinden gelmek için Veri Tabanı kavramı ortaya atılmış ve ilk olarak IBM tarafından kullanılmıştır.

Sonraki gelişmelerle hep yapısal hem de içerik olarak oldukça gelişen veri tabanları aynı zamanda yazılım dünyasının her alanında kullanılmaya başlandı. Aslında bu kullanımda veri tabanlarının standart bir yapısal dile sahip olmaları da önemli bir etkiye sahiptir.

Veri tabanlarının standart sorgulama dili veriye erişimde büyük kolaylık, esneklik ve güvenliği de beraberinde getirdiğinden her programlama dili tarafından desteklenen bir özellik olmuştur. Programlama dillerinin neredeyse tümünde standart veri tabanı sorgulama dili olan SQL için destek sağlanmaktadır.

Bilginin işletmelerin sermayesi olarak kabul edildiği çağımızda, bu bilgileri korumak, gerektiğinde kullanmak, kullanıcıları yetkilendirmek ve içindeki anlamlı bilgileri alacak veri depolama sistemleri işletmelerde olduğu gibi artık her alanda kullanılmaktadır. İşletme ve kuruluşlar için veri tabanlarında tutulan verilerde yeni yaklaşımlara göre sermaye olarak kabul edilmektedir. Bazen bu tür verilerin alınması için çok yüklü miktarda ücretler ödenebilmektedir.

VERİ TABANI İLE DOSYALARIN KARŞILAŞTIRILMASI

Veri tabanları bu kadar yaygın kullanılmadan önce dosyalama sistemleri kullanılırdı ve saklanacak ve işlenecek her tür veri de bu dosyalar üzerinde saklanırdı. Geleneksel dosyalama sistemleri çoğunlukla ücretsiz ve her programcı bu sistemleri tasarlayarak kullanabilirdi. Bu yönleri ile avantajlı gibi görünen geleneksel dosyalama sistemleri verileri artması ve internetin yaygınlaşması ile sorunların da ortaya çıkmasına neden olmuştur. Veri tabanı (Database) ile geleneksel dosyalama sistemlerini karşılaştırmak için iki veri saklama sisteminin iyi ve kötü yönlerini saymak yeterli olacaktır (Özseven, 2012).

- **Uygulama Bağımsızlığı:** Dosyalama sistemleri her bir uygulama için ayrı ayrı tasarlandığından, aynı verilerin her bir uygulamada yeniden saklanması gerekebilmektedir. Oysaki, uygulama bağımsız veri tabanlarını farklı uygulamalar kullanabildiğinden gereksiz veri tekrarları ortadan kalkmaktadır.
- **Veri Yapısı:** Geleneksel dosyalama sistemlerinde verinin yapısını bilmek için uygulamayı geliştiren programcı olmak gerekirken veri tabanlarında



Dosyalama sistemlerinde verilerin yönetilmesi çok zor ve hatalara neden olabilmektedir.

sadece kullanılacak verileri barındıran veri tabanına erişim yetkiniz olması yeterlidir.

- **Veri Güvenliği:** Dosyalama sistemlerinde verilerin güvenliğini sağlamak programı geliştiren programcının yeteneği ile sınırlı kalmaktayken veri tabanı sistemlerinde daha gelişkin bir güvenlik sistemi kullanılmaktadır.
- **Programlama Dili Bağımsızlığı:** Dosyalama sistemlerinde her veriye erişim için farklı diller kullanılması gerekmekte iken veri tabanlarında standart bir sorgulama dili kullanılarak her programcının kullanabileceği bir yapı oluşturulmuştur.
- **Erişim Hızı:** Dosyalama sistemlerinde verilere erişim hızı veri tabanlarına göre çok yavaştır.
- **Kullanıcı Yetkilendirme:** Dosyalama sistemlerinde birden çok kullanıcının verilere erişim hak ve yetkilerini düzenlemek çok büyük çaba gerektirirken veri tabanlarında bu işlemler çok kolaylıkla yapılabilmektedir.

VERİ TABANI TANIMI

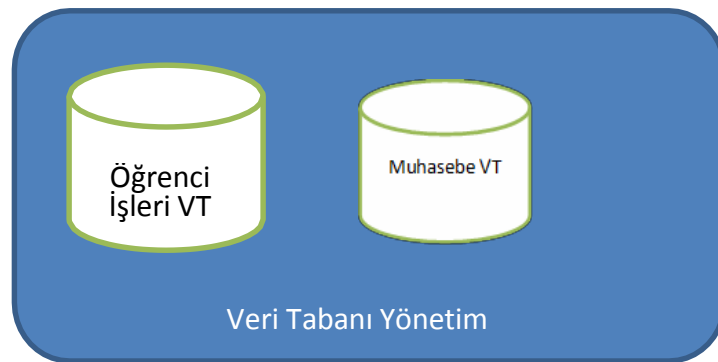
Veri tabanını (*Database*) tanımlamak için birbiri ile ilgili verilerin bir arada tutulduğu sistematik yapılar ifadesi kullanılmaktadır. Aynı zamanda belli bir konuya yönelik oluşturulmuş ve bu alandaki verilerin saklandığı, güncellendiği ve erişim için imkân sağlandığı organizasyonel yapılar şeklinde de tanımlanabilir.

Veri tabanları (VT) günümüzde çok kullanılmalarına karşın eğer yapılandırılması ve planlaması iyi yapılmazsa performans sorunları nedeniyle sistemlerin kilitlenmesi sebep olabilmektedir. Performans sorunlarının en aza indirgenmesi için çeşitli **normalizasyon** teknikleri geliştirilmiştir. Bu teknikler klasik dosyalama sistemlerinden veri tabanlarına geçişte oldukça yarar sağlamıştır.

Günümüzde doğrudan veri tabanı ile veri yönetme yazılımları geliştirenler açısından bu teknikler bilindiğinden performans sorunları çok fazla yaşanmaktadır.



Birbiri ile ilişkili verilerin bir arada tutulduğu yapılara veri tabanı denir.



Şekil 1.1. Örnek Veri Tabanı Gösterimleri

Veri tabanları genel amaçlı olarak oluşturulabileceği gibi sadece konu bazlı da oluşturulabilir. Örneğin bir işletmeye ait genel bir veri tabanı olabileceği gibi **insan kaynakları, muhasebe ve öğrenci işleri gibi özel konulu veri tabanları tasarlanabilir.**

Veri Tabanı Yönetim Sistemleri

Veri tabanı kavramı Veri Tabanı Yönetim Sistemi (*Database Management System*) ile karıştırılmasına karşın aslında farklı kavramlardır. İki kavramı en iyi anlamak için şu tanım daha açıklayıcı olacaktır. Farklı türde uygulamalar ve amaçlar için oluşturulmuş birden çok veri tabanının kullanılmasına ilişkin hak ve yetkiler ile verilerin bütünlüğünü ve güvenliğini sağlamaya yönelik yazılımlara Veri Tabanı Yönetim Sistemleri denir. Kullanılan bilgisayar ve hard disk kapasitesine göre çok sayıda veri tabanını yönetebilir.

Veri Tabanı Yönetim Sistemlerinin (VTYS) günümüzdeki örnekleri Oracle, Microsoft SQL, MySQL ve Sybase sayılabilir. Bu yazılımlardan ilk ikisi çok büyük veri tabanlarını yönetmek için tercih edilen ve ücretli yazılımlar iken MySQL açık kaynak kodlu ve amatör kullanımlar için ücretsiz olarak kullanılabilen bir VTYS'dir. Veri tabanına örnek olarak ise Microsoft Office paketlerinde yer alan Access verilebilir. Bu uygulama tek başına bir veri tabanıdır. Aynı zamanda bir VTYS içinde yer alan muhtelif uygulamalara ait (Personel, Öğrenci, Banka gibi) farklı verilerin saklandığı veri yapılarının her biri bir veri tabanıdır. (Kaya, 2007)

VERİ MODELLERİ

Verileri mantıksal düzeyde düzenlemek için kullanılan yapılar, kavramlar ve işlemler topluluğuna veri modeli (data model) denir. Her VTYS belirli bir veri modelini kullanır. Bir VTYS'yi kullanarak oluşturulacak her veri tabanında yer alacak veriler ve veriler arası ilişkiler, mantıksal düzeyde ilgili veri modeline göre düzenlenir; bu veri modeli kullanılarak veri tabanının kavramsal ve dış şemaları oluşturulur.

Bugüne kadar geliştirilmiş olan çok sayıda veri modeli vardır. Ancak geçmişte ve günümüzde yaygın kullanılan veri modellerini 4 grupta toplamak mümkündür: (Özseven, 2012)

- Sıradüzensel Veri Modeli (Hierarchical Data Model)
- Ağ Veri Modeli (Network Data Model)
- İlişkisel Veri Modeli (Relational Data Model)
- Nesneye-Yönelik Veri Modeli (Object-Oriented Data Model)

Bu sıralama aynı zamanda kronolojik bir sıralamadır.

Sıradüzensel (Hiyerarşik) Veri Modeli en eski model olup 1960 ve 1970'li yıllarda çok kullanılmıştır. 1969'da ortaya çıkan Ağ Veri Modeli 1970'li yıllarda ve 1980'li yılların ilk yarısında kullanılmıştır. İlişkisel veri modeli de ilk kez 1969 yılında ortaya atılmış, 1970'li yılların sonunda kullanılmaya başlanmış ve 1985 yılından sonra yaygınlaşmış bir yaklaşımdır. 1990'lı yıllarda yaygın kullanılan VTYS'lerin hemen hemen tümünün ilişkisel tabanlı olduğu söylenebilir.

Nesneye-yönelik veri modeli yaklaşımı ise on yılı aşkın süredir gündemde olan, günümüzde çok yaygın kullanılsa bile, kullanımı giderek yaygınlaşan bir yaklaşımdır. Geçmişte baktığımızda, ilişkisel yaklaşımın kullanılmaya başlanması ile sıradüzensel ve ağ yaklaşımlarının terk edildiği görülmektedir. Buna karşılık



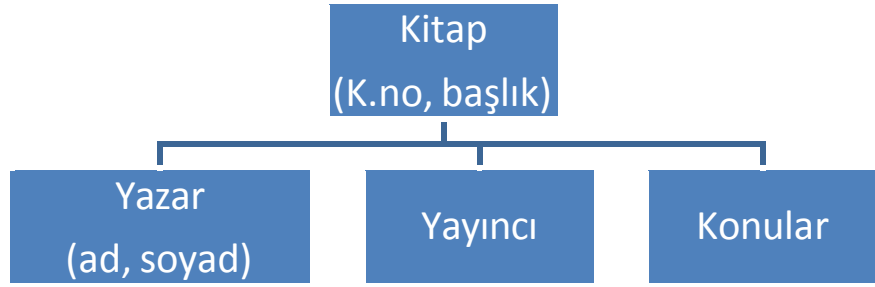
Günümüzde yaygın olarak kullanılan veri tabanı veri modeli, ilişkisel Veri Modelidir.

nesneye-yönelik yaklaşımın kullanılmaya başlanması ile ilişkisel yaklaşım terk edilmemiştir. Günümüzde hem ilişkisel hem de nesneye-yönelik yaklaşımı birlikte kullanan VTYS'lerinin yaygınlaştığı görülmektedir.

Sıradüzensel Veri Modeli (Hierarchical Data Model)

Ağaç veri yapısına benzer. Her kaydın bir ebeveyn kaydı, birçok çocuk kaydı var (IBM IMS: Information Management System).

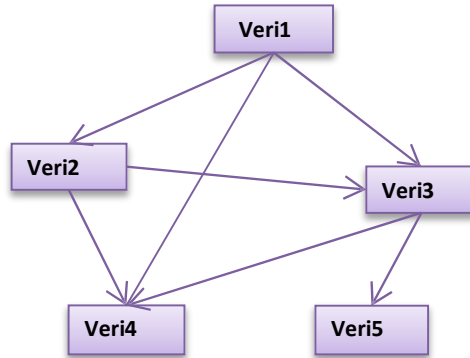
1960'lar ve 1970'lerde veri tabanları için kullanılan ilk modeldir. Bu veri modeli, ana bilgisayar ortamında çalışan yazılımlar tarafından kullanılmaktadır. IBM tarafından çıkarılan IMS (Information Management System) bu türde en çok kullanılan yazılımdır. Sıradüzensel model, bir ağaç yapısına benzer. Model dâhilindeki herhangi bir düğüm, altındaki n sayıda düğüme bağlanırken, kendisinin üstünde ancak bir düğüme bağlanabilir. Sıradüzensel yapının en tepesindeki düğüm noktasına kök denir ve bu düğümün sadece bağımlı düğümleri bulunur. Bu veri yapısını gösteren grafiğe de sıradüzensel tanım ağacı denir. Sıradüzensel bir veri yapısı ve tanım ağacı aşağıda verildiği gibi bir veri modeli oluşturur. Aşağıdaki yapı bir şirketin yönetim kademesinin hiyerarşik yapısını içermektedir.



Şekil 1.2. Sıradüzensel Veri Modeli

Ağ Veri Modeli (Network Data Model)

Ağ veri modeli 1970'li yılların başında geliştirilmiştir. Bir verinin doğası gereği birden çok veri ile ilişkisinin olmasından dolayı hızlıca kabul görmüştür. Bu modelde verilerin birbirine ağ şeklinde bağlandığı varsayılır ve her kaydın bir ebeveyn ve çocukları olduğu varsayılır.



Şekil 1.3. Ağ Veri Modeli

İlişkisel Veri Modeli (Relational Data Model)



Günümüz veri tabanı üreticilerinin tümü İlişkisel Veri Tabanını desteklemektedir.

1970'lerin başında E.F.Codd tarafından geliştirilmiş bir veri modeli şeklini esas alır. Bu sistemde veriler tablolar şeklinde saklanır. Bu veri tabanı yönetim sisteminde; veri alışverişi için özel işlemler kullanılır. Bu işlemlerde tablolar operandlar olarak kullanılır. Tablolar arasındaki matematiksel bağlantılarla (ilişkilerle) temsil edilen ilişkiler belirtilir. Günümüzde hemen hemen tüm veri tabanı yönetim sistemleri ilişkisel veri modelini kullanır. Bu model, matematikteki ilişki teorisine dayanır. İlişkisel veri modelinde (relational data model) veriler basit tablolar hâlinde tutulur. Kayıtlı bu veriler ise anahtar sahalara ile birbiri ile ilişkilendirilerek kullanılır.

Nesneye-Yönelik Veri Modeli (Object-Oriented Data Model)

Günümüzde nesne kavramı her yerde kullanılmaktadır. Pek çok kelime işlemci ve hesap tablosu programlarının alıştığımız görünümüne artık bir de nesnelere eklenmiştir. Ancak bu gerçek anlamda bir nesneye yönelik yazılım demek değildir. Yüzde yüz nesneye yönelik bir yazılımın tamamen nesneye temelli çalışması gerekir. Yazılımın mutlaka nesneye yönelik bir dille yazılmış olması beklenir. Fakat Windows gibi işletim sistemi üzerinde çalışan yazılımlar bu özelliklere tümüyle sahip değildir. Sadece nesne kavramını kullanarak bazı ek özellikler sunar. Nesneye yönelik veri tabanı da, C++ gibi nesneye dayalı bir dille yazılmış olan ve yine C++ gibi nesneye dayalı bir dille kullanılan veri tabanı anlamına gelir. Günümüz teknolojisinde yüzde yüz nesneye yönelik bir veri tabanı yaygın olarak kullanıma sunulmuş değildir. Ancak nesneye yönelik veri tabanlarının bazı üstünlükleri olacağından söz edilmektedir.



Bireysel Etkinlik

- Veri modellerindeki gelişmeler ihtiyaçlardan doğmuştur.
- İlişkisel Veri Modelinden sonra Nesneye Dayalı Veri Modeli neden ortaya çıkmıştır? İnceleyiniz.

VERİ TABANI VERİ TÜRLERİ

Farklı veri tabanı yazılımlarında birbirine benzer veri türleri kullanılsa da bazılarında farklılıklar bulunabilmektedir. Bu farklılıklar veri türüne göre olabileceği gibi bazen de kullanım şekline göre farklılık gösterebilmektedir. Bu kitapta Microsoft Firması tarafından üretilmiş olan Microsoft SQL Sever (MS SQL Server) den bahsedilecektir. (Kaya, 2007)

Bu bölümde anlatılacak veri türlerinin birçoğunun programlama dillerinde de karşılıkları bulunmaktadır. Veri türlerini program içerisinde kullanırken verilerin türlerinin yanı sıra boyutlarının da aynı olmasına özen gösterilmesi veri kaybı ve program hatalarının önüne geçilmesi açısından önemlidir.

MS SQL Server'da veri türleri sütunlar için tanımlanmaktadır ve bu tanımlama verilerimizin saklandığı tablolarda meta veri (meta data) olarak tutulmaktadır. Bir sütun için veri tipi seçilirken bu sütunda ne tür verilerin saklanacağına belirlenmiş olması gerekir. Veriler sadece harflerden, sadece sayılardan veya bunların karışımından oluşabileceği gibi tarih ve daha başka birkaç çeşit veri tipinden oluşabilir.



Metinsel veri tipleri "var" ile başlıyorsa değişken boyutta olduğunu gösterir.

Metin Veri Tipleri

MS SQL Server metinleri kaydedebilmek için 2 farklı temel veri saklama formatına destek verir: ASCII ve Unicode. Ayrıca veri tipinde "var" ifadesi kullanılmış ise bu alanın en fazla olabileceği veri saklama alanının tanımlandığını, az veri girilmesi hâlinde boş alanın veri tabanında açılmayacağını gösterir. Eğer belirtilmemiş ise bu durumda veri alanının belirtilen büyüklükte açılacağını ve değer girilmemişse boş olarak tutulacağını belirtmiş oluruz.

ASCII veri tipleri

ASCII türünden bir sütun tanımlandığında her karakter için 1 Byte veri alanı ayrılır. Parantez içerisinde verilen n değeri bu sütunun alan genişliğini gösterir.

Tablo 1.1. ASCII Türü Metin Veri Tipleri

Tür	Açıklama
char (n)	En fazla 8000 Byte'lık ASCII formatında ve sabit boyutta veri saklamak için kullanılır.
varchar (n)	En fazla 8000 Byte'lık ASCII formatında ve değişken uzunlukta veri saklamak için kullanılır. Standart değerlerden gelmeyen sütunlar için kullanılabilir en etkin veri tipidir.

ASCII Amerikan Standartlar Enstitüsü tarafından kabul edilmiş 0-255 arasında toplam 256 adet standart karakterden ibarettir. Bu kadar sembol de her dildeki harf ve işaretleri ifade etmek için yetersizdir.

Unicode veri tipleri

ASCII kodlama standardı, bütün dillerde yer alan harfleri ve sembolleri ifade etmek için yeterli değildir. Unicode sembolleri 16 Bitlik (2 Bayt) veriler olarak ele alınır. Bu ise $2^{16} = 65536$ adet sembol ifade edebilme kapasitesine ulaşmaktadır. Bu tür veri saklayan veri türlerinin adları 'n' ile başlar (nvarchar ve nchar gibi).

Tablo 1.2. UNICODE Metin Veri tipleri

Tür	Açıklama
nchar (n)	En fazla 8000 Byte'lık UNICODE formatında ve sabit boyutta veri saklamak için kullanılır. Bu ise 4000 karaktere karşı gelmektedir.
nvarchar (n)	En fazla 8000 Byte'lık UNICODE formatında ve değişken uzunlukta veri saklamak için kullanılır. Bu ise 4000 karaktere karşı gelmektedir. Standart değerlerden gelmeyen sütunlar için kullanılabilir en etkin veri tipidir.

Tarih-zaman veri tipleri

Tarih formatı genellikle hassas tarih tutmak ve zamansal sorgulamalar yapmak için tercih edilir. Genellikle bir işin hangi tarihte cereyan ettiğini anlamak

için bu türden sütunlar, GETDATE () fonksiyonu default değer olmak üzere tanımlanır.

SQL Server 2008 tarih ve zaman ile ilgili veri tiplerine, SQL Server adına yeni bir bakış açısı getiriyor. Sağlanan dört yeni veri tipi ve kullanım şekillerini teker teker ele alalım:



Tarih veri tipleri sayısal değerler olsa da tarihlerin sıralanmasında farklı özellikler olduğundan tarih verilerinde kullanılmalıdır.

Tablo 1.3. Tarih Veri Tipleri

Tür	Açıklama
cfetetime	1 Ocak 1753'ten 31 Aralık 9999 arası tarihleri gösterebilen bir veri tipidir. (SQL Server 2005 ve eski sürümlerle aynı tip-Eski sistem uyumu için)
smalldatetime	1 Ocak 1900 ile 6 Haziran 2079 arası tarihleri gösterebilen bir veri tipidir. Dakikanın önemli olmadığı tarihleri kaydetmek için kullanılabilir. (SQL Server 2005 ve eski sürümlerle aynı tip -Eski sistem uyumu için)
date	Gün ay ve yıldan ibaret bir günü belirtebilir. Zaman belirtemez.
time	Tarih hakkında bir bilgi içermez. Sadece günün hangi anı olduğuna dair bilgi tutabilmektedir.
datetime2	İşlev olarak eski klasik DateTime (SQL Server 2005 ve aşağısı) veri tipine karşılık gelmekte olup hem tarih hem de zamanı bir arada tutabilir. Eski tipe göre daha hassas (nanosaniye türünde) tarih zaman tutabilir.
datetimeoffset	DATETIME2 tipi ile aynı özellikleri taşımanın yanı sıra tarih ve zamanın dünya üzerindeki hangi coğrafi zaman dilimi esas alınarak verildiğini de tutabilmektedir.

Sayısal veri türleri

Sayısal verilerin saklandığı veri türleridir. Sayılarda çok çeşitli veriler olduğu gibi veri türlerinde de farklılıklar vardır. Her sayının veri tabanında saklanması farklı boyutta alan tuttuğundan sayısal değerleri saklamak için kullanılacak veri türlerinin doğru bir şekilde seçilmesi hem veri tabanının performansını hem de büyüklüğünü etkilemektedir.

Tablo 1.4. Sayısal Veri Türleri

Veri tipi	Açıklama
Int	Tam sayı girişi yapılabilen veri tipi Minimum Değeri: -2,147,483,647 Maksimum Değeri: +2,147,483,647 Kapladığı Yer 4 byte
Tinyint	Int veri tipinin daha dar bir nümerik aralığa sahip hâli. Sadece pozitif sayılar alabilir. Minimum Değeri: 0 Maksimum Değeri: 255 Kapladığı Yer 1 Byte
Smallint	Int veri tipinin daha dar bir nümerik aralığa sahip hâli. Minimum Değeri: -32,768 Maksimum Değeri: 32,767 Kapladığı Yer 2 byte
Decimal	Ondalıklı rakam girişi yapılabilen veri tipi. Numeric'le aynı görevi görür. Decimal (Precision, Scale) olarak tanımlanır. Precision: Rakamın ondalık dahil kaç haneden oluşacağını belirler. Scale ise ondalık



Sayısal değerler ikili sayı karşılıkları ile saklandığından metinlere göre daha az yer kaplar.

	kısımının kaç haneden oluşacağını belirler. Decimal(3,1) dediğimiz zaman 22,1, -22,1 gibi veri girişleri yapabiliriz. Eğer 22, 12 yazarsak SQL Server bunu 22,1'e dönüştürecektir.
Float	Ondalıklı rakam girişi yapılabilen veri tipidir. Minimum Değeri: -1.79E + 308 Maksimum Değeri: 1.79E + 308 Kapladığı Yer 8 Byte
Real	Real sayı girilebilen veri tipidir. Minimum Değeri: -3.40E + 38 Maksimum Değeri: 3.40E + 38 Kapladığı Yer 4 byte

Bireysel Etkinlik



- Buraya kadar anlatılan veri türlerinden farklı olarak kullanılan diğer veri türlerini araştırınız.
- Bu verilerin hangi amaçla kullanılabileceklerini tartışınız.

VERİ TABANI TABLOLARI

Veri tabanlarında veriler, tablo (table) isimli yapılarda saklanmaktadır.

Tablolar aynı tür verilerin bir arada tutulduğu mantıksal yapılar olarak tanımlanabilir. Tablolar verileri saklamanın yanında verilere ait birçok özelliği de barındırdıklarından veri tabanlarının kullanım alanları çok geniştir.

Tablolar satır ve sütunlardan oluşan veri yapıları olarak gösterildiklerinden satır (row) ve sütun (column) kavramlarının iyice kavranması gerekmektedir.

Tablolarda satır olarak isimlendirilen yapı verinin birbiriyle ilişkili olan farklı türdeki değerlerinin saklandığı yapıdır. Satırlar aynı zamanda kayıt (record) olarak da isimlendirilir. Tablolarda kayıtlar eklenecek her bir veri ile arttığından tablo büyüklüğünü doğrudan belirler.

Tablolardaki diğer önemli bir kavram ise sütunlardır. Sütunlar aynı tipte verilerin kaydedildiği yapılardır. Sütunlar ise alan (field) olarak da isimlendirilmektedir. Alanlar tanımlanırken veri türü dâhil birçok özelliği de tanımlanmaktadır.

Tablolarda kayıtlar programlar ile girilirken alanlar daha çok veri tabanı yönetim sistemi ile yapılmaktadır. Tabloların alanlarına yapılacak ilave ve düzenlemeler verileri doğrudan etkilediğinden her değişiklik yapılamayabilir.

Tablo 1.5. Tablo Örneği

Sicil	Adı	Soyadı	Birimi
125	Ahmet	Narin	Satış
250	Mehmet	Okur	Pazarlama
300	Ayşe	Konak	Muhasebe
...



Tablolar da veri girişi çoğunlukla kayıt olarak yapılır.

Satır ve Sütun Kavramı

Veri tabanı tabloları mantıksal yapılar olmalarına karşın bazı kavramlarla anlam kazanabilmektedir. Bu kavramlardan en önemlileri satır ve sütundur.

Satırlar (Row) için *kayıt (Record)* kavramı da kullanılabilir. *Sütunlar (Column)* için de *alan (Field)* kavramı kullanılabilir.

Bir varlığa ait verilerin bir arada tutulduğu mantıksal yapıya satır denir. Satırlardaki verilerin her biri farklı bir türde olabileceğinden farklı verilerin bir arada tutulduğu veriler bütünü olarak da tanımlanmaktadır. Yukarıda örnek olarak verilen Tablo-5'te yer alan *Ahmet Narin*'e ait verilerin tümü tablodaki sütun tanımına karşılığdır.

Sütun tanımında ise bir tabloda yer alan verilerin aynı tür olanların tutulduğu mantıksal yapıya denir. Yukarıda örnek olarak verilen Tablo- 5'te yer alan Sicili, Adı, Soyadı, Birimi olarak isimlendirilen alanlar sütun olarak tanımlanmaktadır. Sütunlarda saklanacak veri türleri tablo tanımlanırken belirlendiğinde her bir alana ancak belirlenen türde veriler saklanabilir. Sütunların satırlardan farkı her bir sütunun bir isme ve başka özelliklere sahip olabileceğidir. Sütunların özelliklerine ait bazı kavramlar ise aşağıda açıklanacaktır.

Veri Tipleri

Her bir sütun için bir veri türü tanımlanması gerekmektedir. Veri türleri daha önce anlatılan veri türlerinden biri olabileceği gibi veri tabanı yazılımları günün ihtiyaçlarına göre yeni veri türleri de duyurmaktadır. Bu veri türlerinin her biri saklanacak veriye uygun olarak veri tabanı tasarımcısı tarafından belirlenerek veri tabanı mantıksal yapısına dâhil edilmektedir.

Veri tipleri seçilirken bu alana saklanacak veriye uygun alan seçilmesi çok önemlidir.



Örnek

- Örneğin bir alanda sayısal veri saklanacaksa bu alanın sayısal tipte tanımlanması veri tabanının bilgisayarda daha az yer kaplamasını sağlayacağı gibi aynı zamanda bu veriler üzerinden matematiksel işlemler yapılmasına da imkân verecektir.
- Bu alanda saklanacak 12550 sayısını sayısal alanda saklamamız hâlinde 2 Byte'lık bir alana ihtiyaç duyacakken aynı veriyi metin olarak saklayacak olsak her bir hane için 2 Byte'ta saklamak gerektiğinden 10 Byte'lık bir alana ihtiyaç duyacaktır.
- Ayrıca metin olarak veri tabanına saklanan değerlerin matematiksel işlemlere tabi tutulabilmesi için öncelikle sayısal veriye dönüştürülmeleri gerekir, aksi hâlde matematiksel işlemler yapılamaz.

Anahtar Saha

Tablolardaki kayıtlara doğrudan erişmek için veya birçok tabloda yer alan aynı varlığa ait kayıtlar arasında ilişkiler kurmak üzere tanımlanan özel alanlar kullanılır. Bu alanlar bir sütun olabileceği gibi birden fazla sütunun birleşiminde oluşturulabilir. Bu anahtarlar veri tabanı tasarımcısının veya programcının oluşturduğu yapıya göre kendilerinin belirleyebileceği tanımlardır. Veri tabanlarında genel olarak iki tür anahtardan bahsedebiliriz. Ayrıca anahtar saha olmayıp anahtar saha özelliği gösteren sahalarda tanımlayabiliriz.

Primary key

Bunlardan ilki *Primary Key (Birincil Anahtar)* denilen tanımdır. Birincil anahtar bir tabloda sadece bir tane olabilir ve içindeki bilgiler Unique (Tekil) olmak zorundadır. Yani bu alandaki bilgiler tekrar edemez ve yazılan bir bilgi başka bir kayıttaki tekrara yazılamaz. Bu alan eğer birden fazla alanın birleşimi ile oluşturulmuşsa aynı kurallar burada da geçerlidir. Bu sahalarda NULL değer

olamaz yani kayıt oluşturulurken mutlaka bir değer girilmesi gereklidir. Bazı veri tabanı yazılımlarında Primary Key otomatik oluşturulacak şekilde de tanımlanabilir. Bu durumda veri tabanı yönetim sistemi bu alanı belli bir değerden başlatır ve artırarak tanımlar.

Bu anahtar tablolarda kayıt silme ve düzeltme işlemlerinde büyük kolaylık sağlar. Bu anahtar kullanılarak doğrudan ilgili kayda ulaşım sağlanabilir.

Personel Tablosu

Sicil	AdıSoyadı	Telefon	Satış	Amiri
125	Ali Okur	2345	10500	1
150	Veli Yazar	2445	12500	125
160	Ayşe Oku	3456	2000	125
180	Fatma Yaz	2367	3000	150

Satışlar Tablosu

İşlem	ÜrünNo	Adet	Tutar	Satıcı
1	100.010	5	500	125
2	100.011	10	120	125
3	200.100	6	300	180
4	200.090	7	140	150

Ürünler Tablosu

ÜrünNo	ÜrünAdı	Adet	A.Fiyat	S.Fiyat
100.010	Kırmızı Kalem	1000	0,50	0,70
100.011	Mavi Kalem	980	0,50	0,70
...
200.100	Kareli Defter	555	2,50	3,50

Şekil 1.4. Tablolarda Anahtar Saha



Örnek

- Şekil 1’de verilen tablolarda “Sicil”, “İşlem” ve “Ürün No” Birincil Anahtar saha olarak tanımlanmıştır.
- Bu alanlarda aynı veriler kesinlikle tekrar edemez.
- Yine bu tablolardan görüleceği gibi Personel Tablosu ve Ürünler tablosunda Birincil Anahtar saha değerleri rastgele verilmişken Satışlar Tablosundaki Birincil Anahtar saha değeri 1’den başlayarak artımlı olarak kullanılmıştır.
- Bu değerler Veri Tabanı Yönetim Sistemi tarafından verilebileceği gibi programcı tarafından da kontrollü olarak verilebilir.



UNIQUE olarak tanımlanmış bir alana bir veri sadece 1 kez kaydedilebilir.

Unique key

Bu tam olarak bir anahtar olmayıp belli sütunların tekil veri barındırabilmesi için yapılan tanımdır. Böylece kayıtlarda hatalı veri ve tekrarlı veri girilmesi önlenmiş olmaktadır. Bu alana örnek olarak ise öğrenci kayıtlarının tutulduğu veri tabanına öğrenci cep telefonlarının tutulduğu alanın Unique olarak tanımlanması verilebilir. Bu sayede bir cep telefonu numarası sadece 1 kez veri tabanına kaydedilebilir. Birincil anahtarda NULL değeri bulunamazken bu alanda bulunabilmektedir.

Zorlayıcı (Constraint)

Herhangi bir alan için girilebilecek verileri kısıtlayıcı kurallara zorlayıcılar denir. İlgili alana girilebilecek değerleri sınırlayan bir deyim yazılarak tanımlanabilir. Kullanımı bazı veri girişi işlemlerinde faydalıdır ve özellikle yanlış bilgi girişini engelleyerek verilerin doğru girilmesini zorunlu hâle getirir. Kullanıcı, zorlayıcıda belirtilen kural dışında bir veriyi tabloya yazmaya çalıştığında, VTYS hata verir. Böylelikle veri tabanına kullanıcının yanlış değerler girmesi önlenmiş olur ve veri tabanında tutarlılık sağlanmış olur.



İl plaka kodlarının girildiği bir alan için 1 ile 81 arası Constraint tanımlanabilir.



Örnek

- Örneğin, bir öğrencinin sınıf bilgisine ait değerler yazılırken bu alan için rakamsal 1 ile 6 arasında bir zorlayıcı değer tanımlanırsa veri girişi sırasında 1 ile 6 arasındaki değer dışında bir değer sınıf bilgisi alanına yazılması engellenmiş olur.
- Dolayısı ile sınıf için yazılmaması gereken bir değer, bilgi girişi başlangıcında kontrol edilmiş olur.

İndeks (Index)

Veri tabanlarında indeks oluşturularak veriler veri tabanındaki kayıtlı oldukları sıradan başka bir sırada gösterilebilir ve tıpkı kütüphanedeki bir kitab-

ulaşmada olduğu gibi istenilen veriye daha kısa sürede ve kolayca ulaşılabilir.

Temelde indekslerin ilişkisel veri tabanında şu üç işlevi vardır:

- Tekil indeksler, veri ilişkilerini ve veri bütünlüğünü sağlayan birincil anahtar alanlar oluşturmada kullanılır.
- İndeks olan alanın değerine göre bir kaydın kayıtlar arasındaki sırasını gösterir.
- Sorguların neticelenme sürelerini kısaltır.

View (Görüntü)

Bazen, tabloları olduklarından farklı gösterecek filtrelere ihtiyaç duyulur. Bu türden işlevler için VIEW kullanılır. VIEW 'ler, saklanmış sorgulardan ibarettir.

Aslında tablo gibi kullanılsa da hâlihazırda böyle bir tablo veri tabanında bulunmaz, sadece view (görüntüsü) bulunur. VIEW'ler şu görevler için kullanılır:

- Kullanıcıların bazı kritik tabloların sadece belli sütunlarını veya satırlarını görmesi istenildiğinde,
- Kullanıcıların, çeşitli birim dönüşümlerinden geçmiş değerler görmeleri gerektiğinde,
- Hâlihazırdaki tablolarda var olan verilerin başka bir tablo formatında sunulması gerektiğinde,
- Çok kompleks sorguları basitleştirmek için kullanılabilir.

Joining (ilişkilendirme)

İki veya daha fazla tabloyu birlikte sorgulama işlemine join ismi verilir. İlişkisel veri tabanının en temelinde birden fazla tablo üstünde birlikte işlem yapabilmek yatar. Bu sayede verilerin tekrarlanması önlenmiş olur ve sonuçta veri yönetimi kolaylaşır.

VERİ TABANI KULLANICILARI

Veri tabanı kullanıcıları birkaç grup altında toplanabilir. Bu kullanıcıların bazıları işletmelerin büyüklüğüne bağlı olarak birleştirilebilmektedir. Bazı durumlarda da aşağıdaki kullanıcılardan birkaçını program hazırlayan programcılar yerine getirmektedir.

Veri Tabanı Yöneticisi

Veri Tabanı yöneticisi veri tabanındaki en yetkili kullanıcıdır. Genellikle İngilizce ismi olan Database Administrator'un baş harfleri olan DBA ismi ile anılır. DBA'nın birçok görevi vardır. Bunlar:

- Veri tabanı Tasarımı,
- Performans Analizi,
- Erişim Yetkilerini Düzenleme ve Erişim Sağlama,
- Yedekleme ve Geri Yükleme,
- Veri Bütünlüğü Sağlama,
- Sistem Sürekliliği Sağlama şeklinde özetlenebilir.



Veri Tabanı Yöneticisi çok büyük veri tabanlarını yönetmezse verilerin kullanımında performans sorunları yaşanır.

Uygulama Programcısı

Bu kullanıcıların görevleri ise son kullanıcılara yönelik uygulama yazılım geliřtirmek ve veri iřleme dilini kullanmak řeklinde özetlenebilir.

Sorgu Dili Kullanıcıları

Veri tabanından alınması istenen raporlara iliřkin sorgular hazırlamak ve programlar dıřında yapılacak veri güncellemesi veya düzenlemelerini yapacak sorguları hazırlamak olarak tanımlanabilir. Bu kullanıcılar yapacakları sorguları SQL dili ile hazırlar.

Son Kullanıcılar

Uygulama programcılarını tarafından geliřtirilen yazılımları veri tabanı yöneticisinin yetkilendirdiđi tablolar üzerinde kullanan ve iřlemler yapan kiřiler olarak özetlenebilir.



Özet

•VERİ TABANI İLE DOSYALARIN KARŞILAŞTIRILMASI

- Verilerin bilgisayarlarda saklanma süreci ile başlayan veri saklama ortamlarına bakıldığında, veri tabanlarından önce klasik dosyalama sistemlerinin kullanıldığı görülmektedir. Bu dosyalama sistemlerinin zaman içinde ortaya çıkan kısıtları ve eksiklikleri olduğundan veri tabanı kullanımı hızla yaygınlaşmıştır.
- Klasik dosyalama sistemlerinde ki yazılım bağımlılığı, veri yapısının bilinme zorluğu, verileri güvenliği, veriye erişim hızının düşük olması ve kullanıcı sayısındaki kısıtlar veri tabanlarına geçişi hızlandırmış ve kolaylaştırmıştır.

•VERİ TABANI TANIMI

- Veri tabanları, verilerin saklanmasında en önemli yazılımların başında gelmektedir. Veri tabanları farklı konular için oluşturulduğundan bu farklı veri tabanlarını yönetmek, barındırmak ve güvenliğini sağlamak gibi konularda kolaylık sağlayan Veri Tabanı Yönetim Sistemleri kullanılmaktadır.

•Veri Tabanı Yönetim Sistemleri

- Veri tabanı yönetim sistemleri adından da anlaşılacağı gibi farklı yazılım veya uygulamalara yönelik verileri barındıran veri tabanlarının tek bir yazılımla yönetilmesini sağlayan sistemlerdir.

•VERİ MODELLERİ

- Zaman içinde gelişen teknoloji ve ihtiyaçlara göre çeşitli veri tabanı modelleri kullanıma sunulmuştur. Sıra düzensel veri modeliyle başlayan süreç ağ veri modeli bir süre yoluna devam etmiş ve günümüzde yaygın olarak ilişkisel veri modeli kullanılmaktadır. Ancak programlama dillerinin ve teknolojinin yazılımsal nesnelere kullanım ihtiyaçları ile Nesneye Yönelik Veri Modellerini destekleyen veri tabanı yönetim sistemleri de kullanılmaktadır.

•VERİ TABANI VERİ TÜRLERİ

- Veri tabanlarında bilgisayar yazılımlarının desteklediği veri türlerinin tümü desteklenmektedir. Desteklenmemesi hâlinde programlarda veri tabanı kullanımlarında sorunlar yaşanacağından veri tabanı yönetim sistemi geliştiricileri güncel veri türleri için yazılımlarını yenilemektedir. İlk veri tabanları ASCII gibi 256 karakterlik kodları desteklerken günümüz veri tabanları UNICODE ile on binlerce karakter kodunu destekler hâle gelmiştir.
- Veri tabanlarında kişisel verilerden ölçümlerle elde edilen verilere kadar çeşitli veriler saklanmaktadır. Dolayısıyla da veri tabanlarında ad, soyad gibi metinsel veriler yanında çeşitli ölçümlere ait değerlerin saklandığı sayısal değerlerde saklanmaktadır. Günümüzde ise web adreslerinde XML verilerine kadar farklı türde verilerde saklanmaktadır.

•VERİ TABANI TABLOLARI

- Veri tabanlarının mantıksal yapısında temel veri saklama nesnesi olan tablolar ve bunlara ait tanımlar olan satır, sütun, kayıt, alan, anahtar saha ve diğer kavramların bu ders notunun sonraki bölümlerinin daha iyi anlaşılması açısından önemlidir.

•Anahtar Saha

- Veri tabanı yönetim sistemlerinde verinin hızlı, güvenli ve doğru bir şekilde erişiminde etkin olan anahtar sahalar vardır. Bunlardan primary key bir tablodaki en önemli veri erişim anahtarı olarak kullanılmaktadır.



Özet (devamı)

- Veri tabanlarında tabloların yanında view, index gibi çok sayıda farklı nesne de bulunmaktadır. Bu nesnelerin tümü veri tabanı yönetim sistemi içinde yer alan mantıksal tanımlardır aslında.
- Veri tabanlarında büyük veri yığınlarından işe yarayacak verileri süzmek için view, joining ve index gibi kavramlar kullanılmaktadır. Bu mantıksal yapılar verilerin daha güvenli bir şekilde ve hızlı olarak kullanılabilmesini sağlamaktadır.
- VERİ TABANI KULLANICILARI**
- Veri tabanı yönetim sistemleri veri tabanı ve verilerin güvenliğini sağlamak için de çeşitli yöntemler uygulamaktadır. Bunlardan en önemlisi kullanıcı tanımları ve bu kullanıcılara verilen yetkiler olarak düşünülebilir. Bu yetkiler ile kim hangi veriye, tabloya veya nesneye erişecek veya erişemeyecek türünden çok ayrıntılı yetkilendirmeler yapılabilmektedir.

DEĞERLENDİRME SORULARI

1. Geleneksel dosyalar ile veri tabanlarının karşılaştırılmasında aşağıdakilerden hangisi söylenemez?
 - a) Veri tabanları farklı programlara veri sunabilir.
 - b) Dosya sistemleri her program için ayrı tasarlanır.
 - c) Dosyalama sistemleri daha güvenlidir.
 - d) Veri tabanları veriye daha hızlı erişim sağlar.
 - e) Veri tabanları çok sayıda kullanıcıya hizmet verebilmektedir.
2. Veri tabanı ile veri tabanı yönetim sistemi için aşağıdakilerden hangisi doğrudur?
 - a) Veri tabanı yönetim sistemi birden çok veri tabanı barındırabilir.
 - b) Veri tabanı farklı verileri barındırır.
 - c) Veri tabanı veri tabanı yönetim sistemlerini içerir.
 - d) Veri tabanı yönetim sistemleri en küçük veri tabanıdır.
 - e) Access bir veri tabanı yönetim sistemidir.
3. Aşağıdaki veri modellerinden hangisi ilk veri tabanlarında kullanılmıştır?
 - a) Ağ Veri Modeli
 - b) Hierarchical Data Model
 - c) İlişkisel Veri Modeli
 - d) Object-Oriented Data model
 - e) Relational Data Model
4. Aşağıdaki veri türlerinden hangisi ASCII veri türüdür?
 - a) nchar
 - b) datetime
 - c) Int
 - d) varchar
 - e) real
5. Aşağıdaki veri türlerinden hangisi Byte olarak daha fazla alan kaplar?
 - a) Int
 - b) Tinyint
 - c) Smalint
 - d) Real
 - e) Float
6. Veri tabanında bir öğrenciye ait veriler aşağıdaki yapılardan hangisinde saklanır?
 - a) Tablo
 - b) Sütun
 - c) Alan
 - d) View
 - e) Kayıt

7. Bir tabloda tanımlanan primary key için aşağıdakilerden hangisi yanlıştır?
- Tekrarlı veri içerir.
 - NULL değeri içeremez.
 - Bir tabloda bir alanda olabilir.
 - Unique olmalıdır.
 - Kayıt düzeltme ve silme gibi işlemleri kolaylaştırır.
8. Aşağıdakilerden hangisi tabloların istenilen kısımlarını göstermeye yarayan mantıksal yapılardır?
- Index
 - Primary Key
 - Foreign Key
 - View
 - Constraint
9. Aşağıdakilerden hangisi veri tabanı yöneticisinin görevlerinden biri değildir?
- Veri Tabanı Tasarımı
 - Performans Analizi
 - Uygulama Programı
 - Yedekleme ve Geri Yükleme
 - Veri Bütünlüğü Sağlama
10. Büyük veri tabanı yönetim sistemlerinde en fazla kaç tür kullanıcı bulunabilir?
- 3
 - 4
 - 1
 - 2
 - 5

Cevap Anahtarı

1.c, 2.a, 3.b, 4.d, 5.e, 6.e, 7.a, 8.d, 9.c 10.b

YARARLANILAN KAYNAKLAR

Kaya, Y. ve Tekin, R. (2007). Veri tabanı Uygulamaları. Papatya Yayınları, İstanbul.

Özseven, T. (2012). Veri Tabanı Yönetim Sistemleri-1. Murathan Yayınları, Trabzon.