

HASTANE OTOMASYONU I Ders Notları

www.eminkaya.net

2

Veritabanı ve Veritabanı Yönetim Sistemleri (VTYS)

3

Veritabanı Nedir?

- ▶ Veritabanı en genel tanımıyla, kullanım amacına uygun olarak düzenlenmiş veriler topluluğudur. Günümüzde, birçok alandaki veri işlemlerinde çeşitli Veritabanı Yönetim Sistemi programları yaygın olarak kullanılmaktadır.
- ▶ Hastane Yönetim Sistemi uygulamalarında da, verilerimizi saklamak için Veritabanı uygulamalarına ihtiyaç duyulmaktadır.



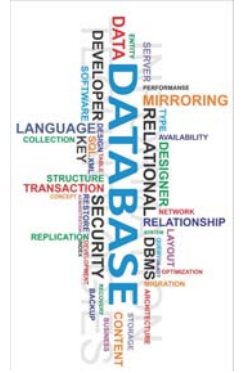
Derleyen: Öğr. Gör. Emin KAYA

www.eminkaya.net

4

Veritabanı

- ▶ Elektronik ortamında saklanan düzenli verilerdir.
- ▶ Sistematik erişim imkanı vardır, yönetilebilir, güncellenebilir, taşınabilir, birbirleri arasında ilişkiler tanımlanabilir.
- ▶ Programlar tarafından işlenebilir.



Derleyen: Öğr. Gör. Emin KAYA

www.eminkaya.net

5

Veritabanlarının Gelişimi

- Veritabanlarına ilişkin ilk çalışmalar **1960'lı** yıllarda başlamış ve ilk veritabanı yönetim sistemi olan IDS (Integrated Data Store) *Charles Bachmann* tarafından geliştirilmiştir.
- **1960 sonrasında** ilk ticari veritabanı yönetim sistemi olan IMS (Information Management System) IBM tarafından piyasaya sunulmuştur.
- **1970'lerde** *Ted Codd* ilişkisel veri modelini tanımlayarak bu modelde bir veritabanı yönetim sistemi oluşturmuştur.
- **1980'lerde**, ilişkisel Veri tabanlı teknolojisinin gelişmesi ile veritabanı sistemlerinin tasarlanmasını ve yönetimini sağlayan SQL (Structured Query Language) dili oluşturulmuştur.

Derleyen: Öğr. Gör. Emin KAYA

www.eminkaya.net

6

Veritabanlarının Gelişimi

- **1990'lı yıllarda** ilişkisel veritabanı modeli yaygınlaşarak veritabanı yönetim sistemleri oluşturulmuştur.
- Nesnelere yönelik, tündengelimli yeni veri modelleri ortaya konmuş ve **1990'ların sonuna doğru** nesneye yönelik teknoloji ile ilişkisel veritabanı modelleri birleştirilerek veritabanı yönetim sistemlerinde yeni uygulama alanları açılarak online olarak bilgi sistemlerini yönetebilen uygulamalar geliştirilmiştir.
- **Günümüzde** bir çok alandaki veri işlemlerinde pek çok Veri Tabanı Yönetim Sistemleri programları yaygın olarak kullanılmaktadır.

Derleyen: Öğr. Gör. Emin KAYA

www.eminkaya.net

Veritabanı Türleri

Veritabanında bulunan veriler farklı şekillerde (modellerde) işlenebilir.

- Hiyerarşik Veritabanı Modeli
- İlişkisel Veritabanı Modeli
- Nesnesel Veritabanı Modeli



Veritabanı Türleri

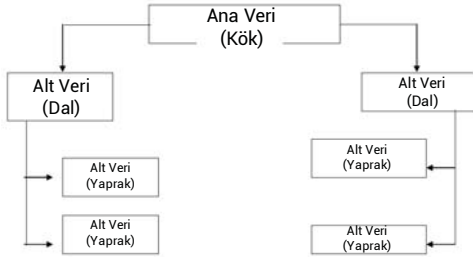
Hiyerarşik Veritabanı Modeli

Bu veritabanı tipi, ana bilgisayar ortamlarında çalışan yazılımlar tarafından kullanılmaktadır. Bu türde en çok kullanılan yazılım, IBM tarafından çıkarılan IMS' dir.

Hiyerarşik veri tabanları, bilgileri bir ağaç (tree) yapısında saklar. Kök (Root) olarak bir kayıt ve bu köke bağlı dal (Branch) ve yaprak (leaf) kayıtlar bu tip veritabanının yapısını oluşturur.

Veritabanı Türleri

Hiyerarşik Veritabanı Modeli



Veritabanı Türleri

İlişkisel Veritabanı Modeli

İlişkisel Veritabanı modellerinde veriler kendi içerisinde ilişkilendirilerek basit tablolar halinde tutulurlar. Burada tablo kümelenirilmiş veri topluluğu anlamına gelmekte olup, satır ve sütunlardan oluşmakta ve veriler bu tablolara kayıt yapısında kaydedilmektedir.

Bir tabloda mevcut sütun bilgilerinin oluşturduğu her satır kayıt olarak adlandırılmaktadır. İlişkisel veritabanları için en az iki tablonun birbirleri ile ilişkilendirilmiş olması gerekmektedir.

Veritabanı Türleri

İlişkisel Veritabanı Modeli

Örneğin hasta takibi amacıyla oluşturulacak bir ilişkisel veritabanında kimlik bilgileri, muayene bilgileri, tahlil sonuçları ayrı ayrı tablolarda tutulmaktadır.

İlişkisel veritabanlarında sorgular yapılarak aşağıdaki işlemler yapılabilir:

- **Kayıt - Sütun Süzme:** Tüm kayıtlar ya da istenen kayıtların elde edilmesi.
- **Tablo Birleştirme:** İki ya da daha çok tablonun birleştirilerek elde edilmesi.
- **Hesaplama:** Alanlar üzerinde hesaplamaların yapılması.
- **Yeni Alanlar:** Yeni alanların oluşturulabilmesi.

Veritabanı Türleri

İlişkisel Veritabanı Modeli

Hasta_ID	Ad_Soyad	Poliklinik_ID	Hekim_ID
112289	Ali CANDAN	4	2
112290	Elif ŞİMŞEK	2	1
112291	Derya ASLAN	3	3
112292	Mesut KALKAN	1	4

ID_No	Poliklinik
1	Dahiliye
2	KBB
3	Ortopedi
4	Nöroloji

ID_No	Hekim_Adi
1	A Hekimi
2	B Hekimi
3	C Hekimi
4	D Hekimi

Veritabanı Türleri

İlişkisel Veritabanı Modeli

İlişkisel veri modeli türlerinden olan E-R (Entitiy-Relationship) modeli günümüzde en yaygın kullanılan veritabanı türüdür. E-R modelinde kullanılan şekiller veritabanının şematik olarak tasarlanmasını sağlar.

Şemalarda kullanılan şekiller ve anlamları şunlardır:



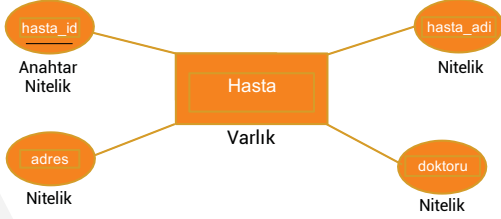
Veritabanı Türleri

İlişkisel Veritabanı Modeli

Varlık; Modelin en temel ögesidir. Kişi, araba, ev veya hasta gibi fiziksel nesnelere olabileceği gibi, şirket, iş veya ders gibi fiziksel olmayan nesnelere de olabilir. Her varlık kendisini tanımlayan, kendisine has niteliklere sahiptir. Birden fazla varlığın oluşturduğu kümeyle varlık kümesi denir. Model içerisinde dikkörtgen ile gösterilir. Varlığın ismi içine yazılır. **Nitelik;** Varlıkların her bir özelliği nitelik olarak ifade edilir. Model içerisinde oval gösterilir. Niteliğin ismi içine yazılır. Nitelik bulunduğu varlığa düz çizgi ile bağlanır. Veritabanında her tablonun bir sütununu ifade eder. Niteliğin değeri her bir varlık için farklıysa **anahtar nitelik** olarak belirlenir. Şema içerisinde altı çizilidir. **Birden fazla anahtar nitelik olamazsa bir tanesi birincil anahtar nitelik olarak tanımlanır.**

Veritabanı Türleri

İlişkisel Veritabanı Modeli



Veritabanı Türleri

İlişkisel Veritabanı Modeli

İlişki; İki veya daha fazla varlık kümesi arasında kurulan anlamlı bağıntılara ilişki denir. İlişkiler E-R şemasında dörtgen ile gösterilir. Dörtgen içine ilişkinin adı yazılır.



Veritabanı Türleri

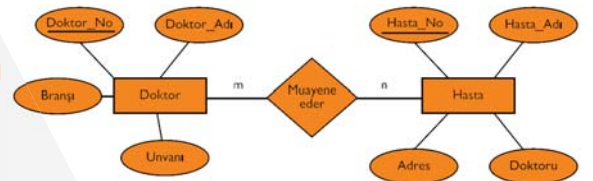
İlişkisel Veritabanı Modeli

İlişki Türleri: Varlık kümeleri aralarında 4 türde ilişki kurulabilir.

- **Bire-Bir (1-1) İlişki:** A varlık kümesinin bir elemanı, B varlık kümesinin sadece bir elemanı ile ilişki kurabiliyorsa bu ilişki türüne bire-bir ilişki denir.
- **Bire-Çok (1-n) İlişki:** A varlık kümesinin bir elemanı B varlık kümesinin birden çok elemanı ile ilişki kurabiliyorsa bu ilişki bire-çok ilişkidir.
- **Çoğu-Bir (n-1) İlişki:** A varlık kümesinin birden fazla elemanı B varlık kümesinin bir elemanı ile ilişki kurabiliyorsa bu ilişki çoğu-bir ilişkidir.
- **Çoğu-Çok (m-n) İlişki:** A varlık kümesinin birden fazla elemanı ile ilişki kurabiliyor ve B varlık kümesinin birden fazla elemanı ile ilişki kurabiliyorsa, bu ilişki türü çoğu-çok ilişki türüdür.

Veritabanı Türleri

İlişkisel Veritabanı Modeli



Nesnel Veritabanı Modeli

Verilerin nesne şeklinde eklenerek modellenen veritabanlarıdır. Karmaşık veriler üzerinde işlem yaparken yüksek performans sunan bir yaklaşımdır. İlişkisel modelin temel özelliklerini taşımakla birlikte nesne yönelimli veri modelinde, bir sorgunun karşılığında mutlaka önceden tanımlanmış belirli bir nesne kümesi olması gerekir. Bir sorgunun sonucu olarak tesadüfi bir nesne kümesinin elde edilmesi mümkün değildir. Çünkü bütün nesnelerin, modelde önceden tanımlanmış olması gerekmektedir.

Nesne veri-tabanı modeli, verilerin herhangi bir noktadan çok kolayca alınabileceği, üç boyutlu bir yapıdan oluşur. İlişkisel veri tabanı verileri iki boyutlu tablolar halinde getirirken, nesne modelinde veriler tek parça olarak gelirler.

Nesnel Veritabanı Modeli

Nesneye Yönelik Veri Modeli'nin İlişkisel Veri Modeline Göre Avantajları

- Veri tipleri esnekler
- Nesne tanımlarında soyutlama yapılabilir
- Veri bütünlüğü daha kolay sağlanır
- Veri yapısında daha fazla genişleme ve yeniden düzenleme imkanı vardır.

Veritabanı tasarımı genel olarak 6 aşamadan oluşur:

I. Gereklik Analizi

Hangi veriler saklanacak?
Ne tür uygulamalar gerçekleştirilecek?
Hangi sorgular kullanılacak?

II. Kavramsal Tasarım (Veri Modellemesi)

Veritabanı hangi verileri içerecek?
Veriler nasıl ilişkilendirilecek?

III. Mantıksal Tasarım

Veritabanı yönetim sistemini tanımlanır.

IV. Şematik Filtreleme

Veritabanındaki veriler arası ilişkileri analiz ederek olası problemleri belirleyerek hazırlanan şematik yapıyı filtreler.

V. Fiziksel Tasarım

Verilerin dosyalanması ve indexlenmesi gerçekleştirilerek veritabanının kavramsal ve mantıksal tasarımda belirlenen gereksinimlere uygunluğu denetlenir.

VI. Uygulama ve Güvenlik Tasarımı

Uygulama programının yazılması, Hangi kullanıcının hangi veriye ulaşabileceğinin belirlenmesi, Belirlenen veriye ulaşım yollarının çalışıp çalışmadığının test edilmesi.

Bir tablo içerisinde yer alacak kayıtların nelerden oluşmasına karar vermeye yarayan düzenlemelere normalizasyon kuralları denilir. Kabul görmüş bazı temel normalizasyon kuralları şunlardır:

1. Bir satırdaki bir alan yalnızca bir tek bilgi içerebilir.
2. Bir tabloda, anahtar olmayan her alan, birincil anahtar olarak tanımlı alanlara bağlı olmak zorundadır.
3. Bir tablo için, anahtarı olmayan bir alan, anahtarı olmayan başka hiç bir alana bağlı olamaz.
4. Tekrarlamaları ortadan kaldırmak için her bir tablonun mümkün olduğunca küçük parçalara bölünmesi gerekir.

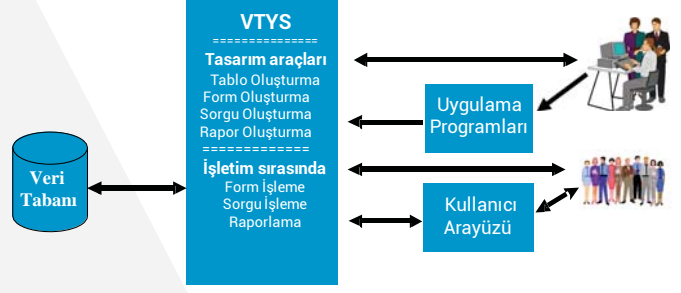
Veritabanı Yönetim Sistemi

Yeni bir veritabanı oluşturmak, kullanıcıların erişimlerini düzenlemek, veritabanını düzenlemek, geliştirmek ve bakımını yapmak gibi çeşitli işlemlerin gerçekleştirildiği bir yazılım sistemidir.

VTYS

VTYS'de verileri tutmak üzere çeşitli türde nesne ve bu nesnelere erişimleri düzenlemek üzere kullanıcılar, roller ve gruplar yer alır. Her bir kullanıcının ulaşabileceği belli hakları vardır. Bu haklar, kısıtlanabilir. Örneğin bir tabloda bir kullanıcı düzeltme yapabilirken, bir başkası sadece verileri sorgulayıp görebilir.

Veritabanı Yönetim Sistemi



Veritabanı Yönetim Sistemi

VTYS

Uygulamadan gelen istekleri alarak bunları veritabanı dosyaları üzerinde, verileri okuyarak ya da yazarak gerçekleştirir. SQL cümlelerini okur ve bu ifadeleri bilgisayarın işletim sisteminin, veritabanı dosyaları üzerinde, verileri okuyacağı ya da yazacağı yönergelerle dönüştürür.

Uygulama Programı

Veritabanı yönetim sistemi ile kullanıcı arasında iletişimi sağlayan bir ya da birden fazla bilgisayar programıdır. Formlar, sorgular ve raporlar oluşturur. Kullanıcıdan verileri alır ya da kullanıcıya verileri gönderir. Kullanıcı davranışlarını veritabanı yönetim sisteminden, veri yönetim etkinliği için isteklere dönüştürür.

Kullanıcı Arayüzü

İşlerini gerçekleştirmek için veritabanını kullanır. Yeni veri ekler, mevcut verileri değiştirir veya verileri siler. Formlar, sorgular ya da raporlar yoluyla verileri okur.

Veritabanı Yönetim Sistemi

VTYS Yararları

- **Veri tekrarı:** Aynı veri farklı kişilerin bilgisayarlarında tekrar tekrar tutulmaz.
- **Veri tutarlılığı:** Aynı verinin değişik yerlerde birkaç kopyasının bulunması "bakım" zorluğu getirir. Bir yerde güncellenen bir adres bilgisi başka yerde güncellenmeden kalabilir ve bu durum veri tutarsızlığına yol açar. VTYS'de veri bir tabloda tutulduğu için veri tutarlılığı sağlanmış olur.
- **Veri paylaşımı:** Veritabanı yönetim sistemi kullanılmadığı durumlarda veriye sıralı erişim yapılır. Yani birden çok kullanıcı aynı anda aynı veriye erişemez. Bir VTYS'de ise verinin tutarlılığını ve bütünlüğünü bozmadan aynı veritabanına saniyede yüzlerce, binlerce erişim yapılabilir.

Veritabanı Yönetim Sistemi

VTYS Yararları

Veri Bütünlüğü: Hasta bilgileri tablosundan bir hasta kaydı silinirse, diğer ilişkili tüm tablolardan o hastanın bilgileri silinmelidir.

Veri Güvenliği: Verinin isteyerek ya da yanlış kullanım sonucu bozulmasını önlemek için çok sıkı sınırlamalar mevcuttur. Veritabanına girmek için kullanıcı adı ve şifreyle korumanın yanı sıra, kişiler sadece kendilerine yetki verilen tabloları ya da tablo içinde belirli sütunları görebilirler.

Veri Bağımsızlığı: Programcı, kullandığı verilerin yapısı, organizasyonu ve yönetimi ile ilgilenmeden veritabanının bunları kendinin koordine etmesi ve yönetmesidir. Veri bağımsızlığı, veritabanı yönetim sistemi programlarının en temel amaç ve özelliklerindedir.

Veritabanı Yönetim Sistemi

VTYS Programları

Kullanılacak veritabanının seçiminde güvenlik, kullanıcı sayısı, aynı zamanda işlem sayısı ve verilerin toplam miktarı önemli rol oynamaktadır. Bu isteklere bağlı olarak uygun bir veritabanı seçimi yapılması gerekmektedir.

Günümüzde en yaygın olarak kullanılan VTYB programları şunlardır:

- Oracle
- DB/2
- MS SQL
- Sybase
- Informix
- Firebird
- My SQL
- MS Access



VTYS Programları

- **Oracle:** Çok büyük ölçekli uygulamalarda tercih edilen bir VTYS'dir.
- **DB/2:** IBM'in geliştirdiği ve büyük ölçekli uygulamalarda tercih edilen bir VTYS'dir.
- **MS SQL:** Microfo'st'un geliştirdiği ve büyük ve orta ölçekli uygulamalarda tercih edilen VTYS™'dir.
- **Sybase:** Büyük ve orta ölçekli VTYS™'dir. Daha çok bankacılık ve kamusal alanlarda tercih edilmektedir.
- **Informix:** Büyük ve orta ölçekli VTYS'dir.
- **Firebird:** Orta ve küçük ölçekli VTYS'dir. Açık kaynak kodludur ve kullanımı giderek yaygınlaşmaktadır.
- **My SQL:** Orta ve küçük ölçekli VTYS'dir. Genellikle Unix-Linux temelli Web uygulamalarında tercih edilmektedir ve açık kodlu bir yazılım olduğu için oldukça popülerdir.
- **MS Access:** Küçük ölçekli ve çoklu kullanıcı desteği olmayan işletim sistemi güvenlik seçeneklerine bağlı bir VTYS'dir.

VTYS Veri Tipleri

Oracle Veri Tipleri

CHAR (sayı): Sabit uzunluktaki alfasayısal verilerin tutulabildiği alanlar için kullanılır. 2000 karakter uzunluğundadır. Eğer, sayı ile ifade edilen numaradan daha kısa uzunlukta veriler girilirse Oracle kaydın sonuna boşluk ekleyerek sabit uzunluğa kadar getirir. Örnek char(20).

VARCHAR2 (sayı): Değişken uzunluklu alfasayısal verilerin tutulduğu alanlar için kullanılır. 4000 karakter uzunluğunda bilgi girilebilir. Örnek varchar2(30).

NUMBER (n, p): Sayılar tam kısım (n) ve ondalıklı kısım (p) olarak saklanmaktadır.

DATE: Tarih tutan alanlar için kullanılır. Bu tip alanlarda, tarih bilgileri ve saat bilgileri tutulabilir.

LONG: 2 GB'a kadar bilgi tutabilen karakter alanlar için kullanılır. Bir tabloda bu tipten ancak bir adet alan tanımlanabilir.

SQL (Structured Query Language)

Açılımı "Yapısal Sorgulama Dili" olan SQL, veritabanı işlemleri ile ilgili komutlardan oluşan bir dildir. Bu dil ile veritabanı üzerinde; veritabanının kendisini oluşturmak, tablo, kullanıcı oluşturmak gibi komutlar ve kayıt ekleme, düzeltme, silme gibi işlemler yapılabilir.

SQL Temelde 4 alt ifade grubundan oluşur.

SQL (Structured Query Language)

1. Veri Tanımlama Dili (Data Definition Language)

Bu gruptaki komutlar kullanılarak, tablo, view (görünüm) gibi veritabanı nesnelere tanımlanır.

2. Veri İşleme Dili (Data Manipulation Language)

Veri üstünde düzenlemeler yapılır.

3. Veri Sorgulama Dili (Data Query Language)

Girilmiş olan veriler üzerinde sorgulamalar yaparak, istenilene ulaşmak amacıyla kullanılır.

4. Veri Kontrol Dili (Data Control Language)

Roller ve kullanıcılar için ifade ve nesne kullanma izni tanımlar.

SQL Komutları

CREATE Komutu

Tablo, view, vb. veritabanı nesnelere oluşturulmada kullanılan komuttur. SQL komutları ile veritabanında işlem yapılabilmesi için, kullanılacak tablolar tanımlanmalıdır.

ALTER TABLE Komutu

Daha önceden oluşturulmuş bir tablonun yapısını değiştirmek için kullanılır

DROP TABLE Komutu

Bir Tabloyu silmek için DROP deyimini kullanılır.

SQL Komutları

DESCRIBE Komutu

Bir tablonun yapısını görmek için DESC[RIIBE] komutu kullanılır. Bu komut ile tanımlanan tabloların alan bilgileri alınır.

INSERT Komutu

Bir tabloya bilgi eklemek için kullanılan komuttur.

SELECT Komutu

Veri tabanındaki tablo veya tablolardan istenilen özellikteki verileri seçip listeleme için kullanılır.

SQL
Komutları**UPDATE Komutu**

Tablodaki bir kayıt veya kayıtlar güncellenmek istenirse UPDATE komutu kullanılır.

DELETE Komutu

Tablodaki bir kayıt veya kayıtlar silinmek istenirse DELETE komutu kullanılır.

SUM Fonksiyonu

Belirtilen sütundaki bütün değerlerin toplamını alır.

AVG Fonksiyonu

Belirtilen sütundaki bütün değerlerin ortalamasını alır.

SQL
Komutları**MAX Fonksiyonu**

Belirtilen sütundaki en büyük değeri verir.

MIN Fonksiyonu

Belirtilen sütundaki en küçük değeri verir.

COUNT Fonksiyonu

Tablo içinde, her hangi bir sayma işlemi gerçekleştirmek için kullanılır.

CREATE USER Komutu

Kullanıcı oluşturmayı sağlar. Kullanıcı veritabanı nesnelere sahiptir.

SQL
Komutları**CREATE ROLE Komutu**

Rol oluşturmayı sağlar. Rol veritabanındaki hakların toplu haline denir. Kullanıcı rolleri kullanılarak sistemin güvenliğini daha kolay sağlayabilir.

GRANT Komutu

Sistem ya da nesne haklarının kullanıcılara veya rollere atanması için kullanılan komuttur.

REVOKE Komutu

Sistem ya da nesne haklarının kullanıcılardan veya rollerden geri alınması için kullanılan komuttur.