

Veri Tabanı

Veri Tabanına Giriş

Hüseyin Ahmetoğlu

Veri Tabanı Nedir?

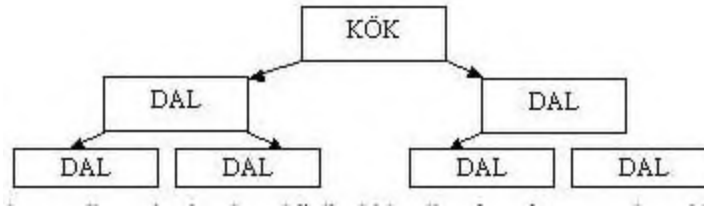
- ▶ Birbiriyle ilişkisi olan verilerin tutulduğu, kullanım amacına uygun olarak düzenlenmiş veriler topluluğunun, mantıksal ve fiziksel olarak tanımlarının bulunduğu bilgi depolarıdır.
- ▶ ÖRNEKLER:
 - Üniversite- Öğrenci İşleri Bilgi Sistemi
 - Hastane-Hasta, doktor, tedavi, araç-gereç, mali bilgiler
 - Ticari bir şirket- Müsteri, Ürün, Satış, Ödeme, Teslimat bilgileri
 - Banka-Müşteri, mevduat, kredi kartı, kredi bilgileri
- ▶ Basit bir web uygulamasından uluslararası kuruluşların büyük ve karışık verilerine kadar pek çok alanda veri tabanı uygulamalarına ihtiyaç duyulmaktadır.

Veri Tabanı Yönetim Sistemi Nedir?

- ▶ Yeni bir veri tabanı oluşturmak, veri tabanını düzenlemek, kullanmak, geliştirmek, bakımını yapmak için çeşitli karmaşık işlemlerin gerçekleştirildiği bir yazılım sistemidir.
- ▶ Veri Modeline Göre
 - Hiyerarşik
 - Ağ
 - İlişkisel
 - Nesneye Yönelik
- ▶ Kullanıcı Sayısına Göre
 - Tek kullanıcı
 - Çok kullanıcı

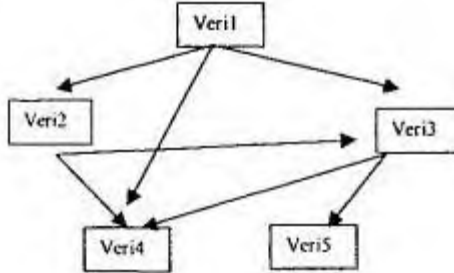
Hiyerarşik veritabanları

- ▶ Veri tabanları için kullanılan ilk modeldir.
- ▶ Hiyerarşik veri tabanları bilgileri bir ağaç yapısında saklarlar.



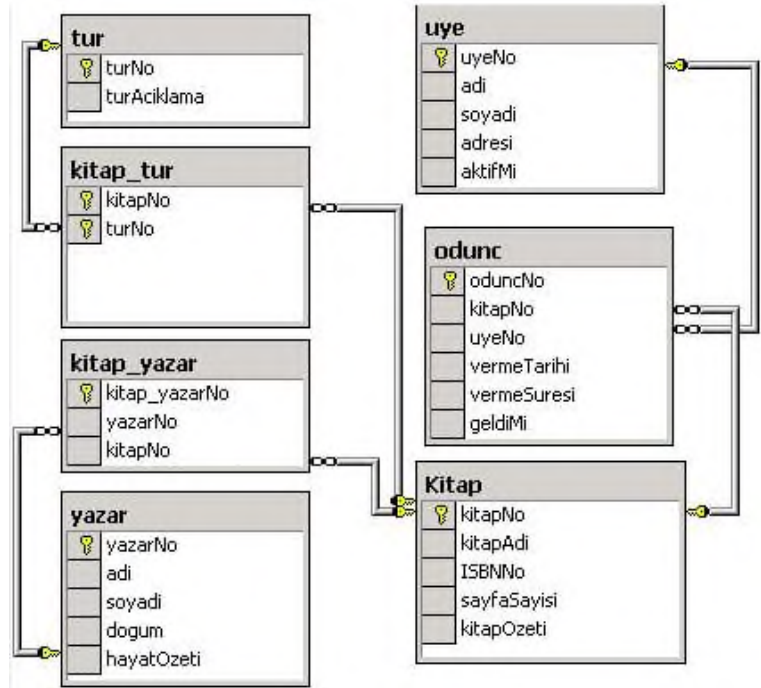
Ağ veritabanları

- Hiyerarşik veritabanları yetersiz kalınca 1960'ların sonunda verilerin ağaçların daha gelişmiş hali olan graflar şeklinde saklandığı yapı ortaya çıkmıştır.



İlişkisel veritabanları

- ▶ 1970'lerin başında geliştirilmiştir.
- ▶ Bu sistemde veriler tablo şeklinde saklanır.
- ▶ Tablolar arasındaki bağlantılar matematiksel ilişkilerle gösterilir.
- ▶ Günümüzdeki hemen hemen tüm veri tabanı programları bu yapıdadır.



Nesneye Yönelik veritabanları

- ▶ Günümüzdeki pek çok kelime işlemci ve hesap tablosu programında kullanılan nesneler artık veritabanlarında da kullanılmaktadır.
- ▶ Nesneye yönelik veritabanı C++ gibi nesneye yönelik bir dille oluşturulan ve yine bu tarz bir dille kullanılan veri tabanı anlamına gelir.

Neden veri tabanı kullanılır?

- ▶ Verilerin tutulması, saklanması ve erişilmesinde geleneksel yaklaşım verilerin ayrı ayrı dosyalarda gruplanması yaklaşımını kullanmaktadır.
- ▶ Verilerin artması, verilere aynı anda erişme ve düzenlenme ihtiyacı ile geleneksel yaklaşım yetersiz kalmıştır.

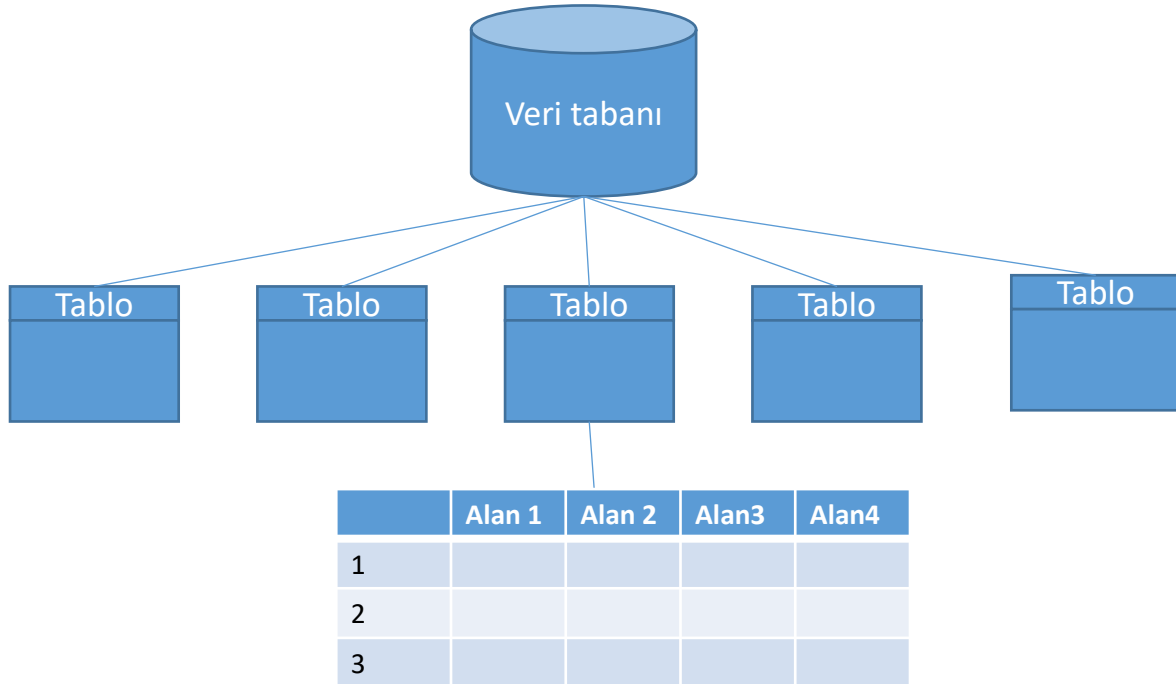
Veri Tabanı Yaklaşımının Avantajları

- ▶ Ortak verilerin tekrarını önlenmesi;
- ▶ Verilerin merkezi denetiminin ve tutarlılığının sağlanması
- ▶ Veri paylaşımının sağlanması
- ▶ Fiziksel yapı ve erişim yöntemi karmaşıklıklarının, çok katmanlı mimarilerle kullanıcıdan gizlenmesi,
- ▶ Her kullanıcıya yalnız ilgilendiği verilerin, alışıktığı kolay, anlaşılır yapılarda sunulması
- ▶ Sunulan çözümleme, tasarım ve geliştirme araçları ile uygulama yazılımı geliştirmenin kolaylaşması.
- ▶ Veri bütünlüğünün gerekli olanakların sağlanması,
- ▶ Güvenlik ve gizliliğin istenilen düzeyde sağlanması
- ▶ Yedekleme, yeniden başlatma, onarma gibi işletim sorunlarına çözüm getirilmesi

Veri Tabanı Yönetim Sistemleri

- Oracle database
- IBM DB/2
- Adaptive Server Enterprise
- Informix
- Microsoft Access
- Microsoft SQL Server
- Microsoft Visual FoxPro
- **MySQL**
- PostgreSQL
- Progress
- SQLite
- Teradata
- CSQL
- OpenLink Virtuoso

Veri Tabanı Yapısı



Tablo

- ▶ Bir veritabanı tablolarda saklanan verilerden oluşur.
- ▶ Tablolar verilerin satırlar ve sütunlar halinde düzenlenmesiyle oluşan veri grubudur.
- ▶ Örneğin ders içeriği ve öğrenci bilgilerini veri tabanında saklamak için 2 tablo oluşturulur:
 - Öğrenci_bilgileri
 - icerik

- ▶ Tablo içindeki her bir bilgi **kayıt**, Sütunlar ise **alan** olarak isimlendirilir.

- ▶ Örneğin öğrenci bilgileri tablosunda

Alan-Sütun

- Öğrenci numarası,
- adı soyadı,
- doğum tarihi,
- doğum yeri,
- e-mail adresi

Ogr_no	Ad_soyad	d_tarih	d_yeri	e-mail
1	Ayşe Öztürk	01.11.1979	Konya	ayse@gazi.edu.tr
2	Sema Özdemir	24.05.1975	Ankara	sema@gazi.edu.tr
3	Serdar Gülpınar	06.06.1983	Adana	serdar@gazi.edu.tr
4	Mehmet Efe	11.02.1978	Niğde	mehmet@gazi.edu.tr
5	Zerrin Polat	22.08.1980	Antalya	zerrin@gazi.edu.tr
6	Ulviye Kubalı	12.12.1984	İstanbul	ulviye@gazi.edu.tr

Kayıt
-
Satır

Veri Türleri

- ▶ Veri tabanında tutulan kayıtların yapısı hakkında bilgi sahibi olmak için alanların bazı özelliklerinin önceden tanımlanması gerekir.
- ▶ Örneğin personel sicil numarası mutlaka tam sayı, ad soyad harflerden oluşması gibi
- ▶ **MYSQL Veri Türleri**
 - Sayısal (Numeric)
 - Tarih Saat (Date and Time)
 - Metinsel (String)
 - Uzamsal (Spatial)

MYSQL Veri Türleri

► TINYINT:

- Çok küçük tam sayı değerler içindir
- Signed tanımlı durumda iken alabileceği değerler -128 ile 127 arasındadır.
- Unsigned tanımlı aralık 0 ile 255 arasındadır.

► SMALLINT:

- Küçük tam sayı değerler içindir
- Signed tanımlı durumda iken alabileceği değerler -32768 ile 32767 arasındadır.
- Unsigned tanımlı aralık 0 ile 65535 arasındadır.

► MEDIUMINT:

- Orta büyüklükteki tam sayı değerler içindir.
- Signed tanımlı durumda iken alabileceği değerler -8388608 ile 8388607 arasındadır.
- Unsigned tanımlı aralık 0 ile 16777215 arasındadır.

► INT(n):Tamsayı

- Normal büyüklükteki tamsayı değerler için.
- Signed tanımlı durumda iken alabileceği değerler -2147483648 ile 2147483647 arasındadır.
- Unsigned tanımlı aralık 0 ile 4294967295 arasındadır.

► BIGINT:

- Büyük tam sayı değerler içindir
- Tam sayı -9223372036854775808'den 9223372036854775807'e

MYSQL Veri Türleri

► FLOAT:

Sayıları kesirleri ile birlikte tutar.

Max. karakter genişliği parametre olarak alınır.

(23 basamağa kadar)

► DOUBLE:

Sayıları kesirleri ile birlikte tutar.

Max. karakter genişliği parametre olarak alınır.

(24'ten 53 basamağa kadar)

► DECIMAL:

Sayıları kesirleri ile birlikte tutar.

Tam kısmı maksimum 64

Kesirli kısmı maksimum 30 değerini alabilir.

MYSQL Veri Türleri

- ▶ **DATETIME:**

Yıl+Ay+Gün+Saat+Dakika+Saniye biçimindeki zaman bilgisi

YYYY-MM-DD HH:MM:SS

- ▶ **TIMESTAMP:**

1 Ocak 1970'den 18 Ocak 2038'e kadar olan ve Yıl+Ay+Gün+Saat+Dakika+Saniye biçimindeki zaman bilgisi.

YYYYMMDDHHMMSS

- ▶ **DATE:**

1000-01-01'den 9999-12-31'e kadar değişebilen tarih alanı.

YYYY-MM-DD

MYSQL Veri Türleri

- ▶ **BOOL:**

0 veya 1 değerini alan veri türüdür.

- ▶ **CHAR(n):**

n sayısı kadar karakteri olan sabit uzunluklu veridir.

- ▶ **TEXT:**

En fazla 65535 karakter alabilen metin alanı.

- ▶ **MEDIUMTEXT:**

En fazla 16777215 karakter alabilen metin alanı

- ▶ **VARCHAR(n):**

n sayısını geçmemek şartıyla değişen boyutta karakter

Anahtar (Key)

- ▶ Anahtar bir veya birden fazla alanın bir satır için niteleyici olarak girilmesi için zorlanan bir çeşit zorlayıcıdır.
- ▶ 2 çeşit anahtar vardır:
 - Birincil Anahtar (Primary Key)
 - Yabancı Anahtar (Foreign Key)

Birincil anahtar

- ▶ Bir kayıta ulaşmayı sağlayacak anahtar veridir.
- ▶ Örneğin öğrenciler arasında iki Ahmet var. Arama yaparken istediğimiz Ahmet'i bulmak için her bir öğrenciye özel bir numara olmalıdır.
- ▶ Örneğin öğrenci numarası
- ▶ Birden fazla alanda birlikte birincil anahtar olabilir

Yabancı anahtar

- ▶ Bir tabloya girilebilecek kayıtları başka bir tablonun belli alanındaki verilerle sınırlandırmaya ve ilişkilendirmeye yarar.
- ▶ Örneğin öğrencilerin not verilerinin girildikleri tablodaki her satıra öğrenci bilgileri tablosundaki öğrenci_no ile eşleşen bir değer girilmesi gibi.

Veri tabanı Tasarlama

1. Nesneler tanımlanır

Kütüphane sistemi: kitap, üyeler, türler, ödünç hareketleri

Veri tabanı Tasarlama

2. Her nesne için bir tablo oluşturulur.

kitap,
uyeler,
turler,
odunc_hareketleri

Veri tabanı Tasarlama

3. Her tablo için bir anahtar alan seçilir

Kitap tablosu: **kitapno**

Üyeler tablosu: **uyeno**

Veri tabanı Tasarlama

4. Nesnelerin her bir özelliği için tabloya sütun eklenir

Kitap tablosu: kitapno, yılı, yazarı, adı, ilgili olduğu alan

Veri tabanı Tasarlama

5. Tekrarlayan nesne özellikleri için ek tablolar oluşturulur.

İstek tablosu

uyeno	İstek_tarihi	Kitap_adi	Kitap_yili	Kitap_yazari	ilgili _alan

Veri tabanı Tasarlama

6. Tablo ile doğrudan ilişkili olmayan alanlar belirlenir.

Ödünç hareketleri tablosunda kitabı ödünç alan üyenin adresi doğrudan bu tablo ile ilişkili değildir.

Bu veri üye bilgilerinin tutulduğu **uyeler** tablosunda yer almalıdır.

Veri tabanı Tasarlama

7. Tablolar arasındaki ilişkiler tanımlanmalıdır.

Tanımlanan tablodaki alanların birbiri ile ilişkisi tanımlanır. Örneğin **uyeler** tablosundaki **uyeno** ile **odunç** tablosundaki **uyeno** alanı ilişkilendirilmelidir.

Normalizasyon -Tanım

- ▶ Normalizasyon; veri tabanı tasarım aşamasında veri tekrarını, veri kaybını veya veri yetersizliğini önlemek için gerçekleştirilen işlemlerdir.
- ▶ Normalizasyon tanım olarak ise ; (Ayrıştırma), veritabanlarında çok fazla sütun ve satırdan oluşan bir tabloyu tekrarlardan arındırmak için daha az satır ve sütun içeren alt kümelerine ayrıştırma işlemidir.
- ▶ Normalizasyon yapılırken uyulması gereken kurulların her birine normal form adı verilir.
 - Birinci Normal Form (1NF)
 - İkinci Normal Form (2NF)
 - Üçüncü Normal Form (3NF)
- ▶ Daha yüksek düzey formlar da vardır.
- ▶ İlk üç düzey ihlal edilirse
 - Kayıt güncelleme
 - Kayıt silme
 - Kayıt bulmada zorluk çekilir.
- ▶ 3NFde olan tablolar 1NF ve 2NFye uygundur. 2NFde olan tablolarda 1NFye uygundur.

Normalizasyon – Giriş

- Normalizasyon yapılırken uyulması gereken kurulların her birine **normal form** adı verilir.
 - Birinci Normal Form (1NF)
 - İkinci Normal Form (2NF)
 - Üçüncü Normal Form (3NF)
- Daha yüksek düzey formlar var ama çok fazla kullanılmıyor.
- İlk üç düzey ihlal edilirse
 - Kayıt güncelleme
 - Kayıt silme
 - Kayıt bulmada zorluk çekilir.
- 3NFde olan tablolar 1NF ve 2NFye uygundur. 2NFde olan tablolarda 1NFye uygundur.

Normalizasyon Amaçları

- ▶ Veri Bütünlüğünün Sağlanması
- ▶ Gereksiz veri tekrarını önleyerek verilerdeki bozulmaları önlemek
- ▶ Uygulamadan Bağımsızlık
- ▶ Uygulama değişse bile veritabanı tutarlı olarak çalışmalı
- ▶ Performansı Arttırmak
- ▶ Veri tekrarı en aza iner ve arama hızlı olur.

Normalizasyon Kuralları

- Birinci Normal Form (First Normal Form) 1NF
- İkinci Normal Form (Second Normal Form) 2NF
- Üçüncü Normal Form (Third Normal Form) 3NF

Birinci Normal Form

- Tekrarlanan sütun yapıları olmamalıdır
- Birden fazla bilgi tek bir sütunda olamaz.
- Bir alan içerisindeki bilgi özel karakterlerle ayrılarak tutulmamalıdır.

Telefon Tablosu	
Ad Soyad	Telefon
Ali Coşkun	3710, 3712
Mehmet Kaya	3714, 3715, 3716
Ahmet Demir	3750

Adres Tablosu		
Ad Soyad	Adres1	Adres2
Ali Coşkun	Ordu	Çivril Atasay Kamer MYO Çivril / Denizli
Mehmet Kaya	Denizli	
Ahmet Demir	Adana	

Birinci Normal Form

- 1Nf uyum hale geldi fakat 2NF ve 3NF ye uygun değil

Telefon Tablosu		
Ad	Soyad	Telefon
Ali	Coşkun	3710
Ali	Coşkun	3712
Mehmet	Kaya	3714
Mehmet	Kaya	3715
Mehmet	Kaya	3716
Ahmet	Demir	3750

Adres Tablosu		
Ad	Soyad	Adres1
Ali	Coşkun	Ordu
Ali	Coşkun	Çivril Atasay Kamer MYO Çivril / Denizli
Mehmet	Kaya	Denizli
Ahmet	Demir	Adana

Birinci Normal Form

Sorunlar

- ▶ Veri Ekleme
 - Eklenen verinin daha önce olup olmadığının kontrolü zor
- ▶ Veri güncelleme
 - Kişiye ait bir telefon numarası güncellenmek istediğine hangisinin güncelleneceği
- ▶ Veri silme
 - Silinecek verinin hangisi olduğu

Telefon Tablosu		
Ad	Soyad	Telefon
Ali	Coşkun	3710
Ali	Coşkun	3712
Mehmet	Kaya	3714
Mehmet	Kaya	3715
Mehmet	Kaya	3716
Ahmet	Demir	3750

İkinci Normal Form

- ▶ Tabloda bir birincil anahtar olmalı ve anahtar olmayan sütunlar birincil anahtara bağımlı olmalı.
- ▶ Birincil anahtar birden fazla sütundan oluşuyorsa tablodaki veriler her iki sütuna da bağımlı olmalıdır.

OgrenciVeNot Tablosu				
<u>ogrNo</u>	Ad	Bolum	<u>DersKodu</u>	Not
759	Ali	Bilgisayar	c121	60
759	Ali	Bilgisayar	c122	70
760	Mehmet	Bilgisayar	c121	65
761	Kemal	Büro	c134	90

Öğrenci bilgileri ve not bilgileri ayrılmalı.

İkinci Normal Form

- İkinci normal forma uygun hali

Ogrenci Tablosu		
<u>ogrNo</u>	Ad	Bolum
759	Ali	Bilgisayar
760	Mehmet	Bilgisayar
761	Kemal	Büro

Not Tablosu		
<u>ogrNo</u>	<u>DersKodu</u>	Not
759	c121	60
759	c122	70
760	c121	65
761	c134	90

Üçüncü Normal Form -1

- Anahtar olmayan sütunlar anahtar sütuna tam bağımlı olmalı. Anahtar olmayan sütuna bağımlı olmamalı.

Ürün Tablosu			
<u>ÜKodu</u>	Ad	Birim	Birim Adı
k001	Alçı	1	Kg
k002	Tel	2	Cm
k003	Boya	3	Lt

Üçüncü Normal Form -2

Ürün Tablosu		
<u>ÜKodu</u>	Ad	Birim
k001	Alçı	1
k002	Tel	2
k003	Boya	3

Birim Tablosu	
Birim	Birim Adı
1	Kg
2	Cm
3	Lt

Normalizasyon Örnek-1

- 1NF Uygun 2NF değil

<u>musteriNo</u>	<u>urun_id</u>	sehir	sehirKodu	miktar
M145	U1	Denizli	1	100
M145	U2	Denizli	1	150
M151	U2	Tokat	2	75
M149	U3	Samsun	3	200
M148	U1	Tokat	2	140

Normalizasyon Örnek 1-devam

- 2NF uygun 3NF uygun değil.

<u>musteriNo</u>	<u>urun_id</u>	miktar
M145	U1	100
M145	U2	150
M151	U2	75
M149	U3	200
M148	U1	140

<u>musteriNo</u>	sehir	sehirKodu
M145	Denizli	1
M151	Tokat	2
M149	Samsun	3
M148	Tokat	2

Normalizasyon Örnek 1-devam

<u>musteri</u>	<u>urun_id</u>	miktar
M145	U1	100
M145	U2	150
M151	U2	75
M149	U3	200
M148	U1	140

<u>musteri</u>	sehirKodu
M145	1
M151	2
M149	3
M148	2

sehirKodu	sehir
1	Denizli
2	Tokat
3	Samsun

KAYNAKLAR

- ▶ Veri Tabanı Yönetim Sistemleri. (2016, March 01). Retrieved from http://sibelsomyurek.com/veritabani/ders_notlari.html
- ▶ Köseoğlu, K. (2005). Veri Tabanı Mantığı. Şefik Matbaası. İstanbul
- ▶ Alokoç Burma, Z. (2005). Veritabanı Yönetim Sistemleri ve SQL / PL - SQL / T – SQL. Seçkin Yayıncılık. Ankara
- ▶ (2021, March 14). Veritabanı -I. Retrieved from <https://www.pau.edu.tr/ali/tr/sayfa/veritabani-i>