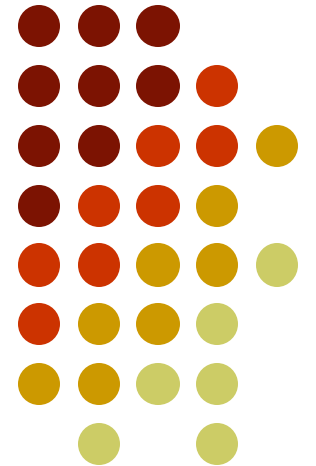
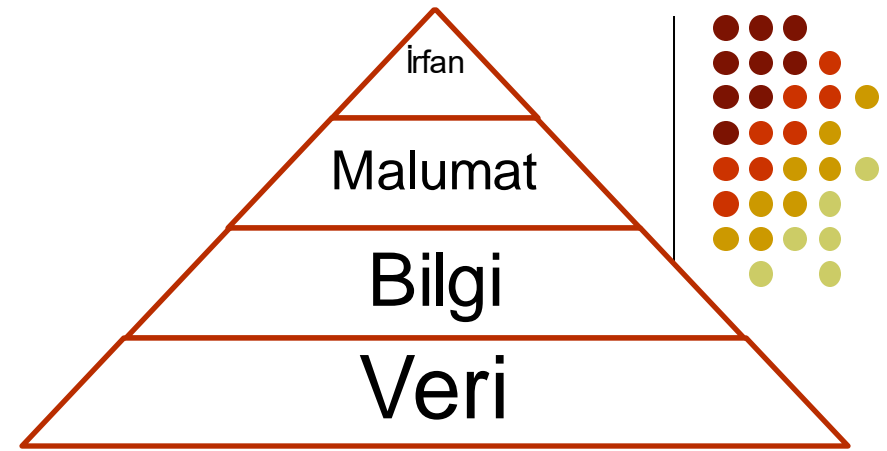


# BİM381

Veritabanı Sistemleri  
M.Zahid Gürbüz



# Veri Piramidi



- Data : Veri
  - ölçüm, deney, gözlem, sayım ya da araştırma yolu ile elde edilen, işlem görmemiş sayılar, yazılar, rakamlar, karakterlerdir.
- Information : Mâlûmat
  - Veri üzerine bir takım işlemler yapılarak verinin anlamlı halidir. Gereksiz ve hatalı bilgileri atma, farklı veri kümelerini birleştirme, toplanan verilerin alakalı veya doğru olmasını doğrulama, verileri görselleştirme gibi işlemler yapılabilir
- Knowledge : Bilgi
  - Malumatı, hedeflerimiz doğrultusunda nasıl kullanılacağını belirlediğimiz halidir.
- Wisdom : İrfan, Bilgelik
  - Bilginin icraata dönüşmüş halidir. Elde edilen bilgilerin aktif olarak kullanıldığı durumdur.

# Veritabanı



- Veritabanı genellikle bir bilgisayar sisteminde elektronik olarak depolanan yapılandırılmış bilgi veya veriden oluşan düzenli bir koleksiyondur.

*Oracle*

- Veritabanları yapılandırılmış bilgi veya verilerin depolandığı alanlardır

*Wikipedi*

# Veritabanı ve Minidünya



- Veri tabanı
  - Mantıksal olarak tutarlı verilerin toplanması yapılır
  - Belirli bir amaç için oluşturulmuştur
- Minidünya (Miniworld)
  - Gerçek dünyanın bir yönünü (özelliğini) temsil eder (Örneğin, bir üniversitedeki öğrenci notları ve transkriptleri.)



# Veritabanı Yönetim Sistemleri (VTYS)

- Veri tabanı yönetim sistemi (VTYS, veri tabanlarını tanımlamak, yaratmak, kullanmak, değiştirmek ve veri tabanı sistemleri ile ilgili her türlü işletimsel gereksinimleri karşılamak için tasarlanmış sistem ve yazılımdır.
  - MySQL, PostgreSQL, Microsoft SQL Server, Oracle Database ve Microsoft Access gösterilebilir.

*Vikipedi*

- Veritabanı genellikle bir veritabanı yönetim sistemi (DBMS) ile kontrol edilir. Veri ve DBMS ve aynı zamanda bunlarla ilişkili uygulama yazılımları bir araya getirildiğinde sıklıkla yalnızca veritabanı olarak kısaltılan veritabanı sistemi olarak ifade edilir
- Kullanıcıların bir veritabanı oluşturmasını ve korumasını sağlar
- Çeşitli kullanıcılar ve uygulamalar arasında veri tabanlarının tanımlanması, yapılandırılması, yönetilmesi ve paylaşılması süreçlerini kolaylaştıran genel amaçlı bir yazılım sistemidir.

# VTYS İşlevleri



- Veri tabanı **tanımlama**
  - Depolanacak verilerin veri türlerini, yapılarını ve kısıtlamalarını belirtir
  - Meta veri: Veri tabanı tanımı veya açıklayıcı bilgi
  - VTYS tarafından bir veritabanı kataloğu veya sözlük biçiminde depolanır
- Veri tabanı **oluşturma** (yükleme)
- Verileri, uygun formatta VTYS tarafından denetlenen depolama ortamlarında **saklama**.

# VTYS İşlevleri



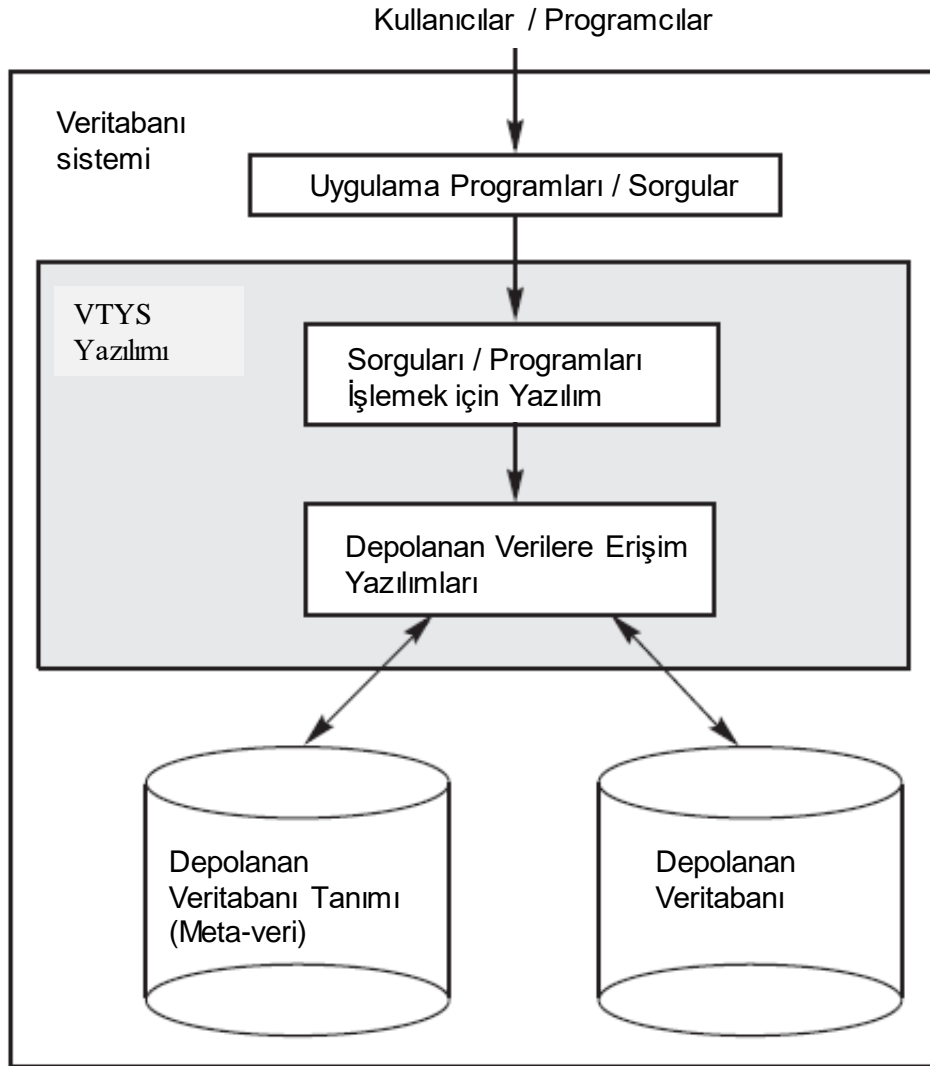
- Veritabanını **değiştirme**
  - Değişiklik(Modification): İçeriğine yapılan eklemeler(insertions), silinmeler(deletions) ve güncellemeler(updates).
  - Alma(Retrieval ): Sorgulama(querying), rapor oluşturma
- Veritabanını **paylaşma**
  - Birden fazla kullanıcının ve programın aynı anda veritabanına erişmesine izin verme

# VTYS İşlevleri

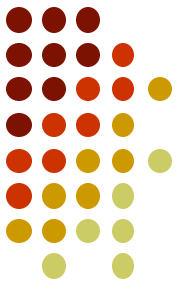


- Veri tabanı sisteminin **Koruma** (Protection )
  - HW veya SW arızalarına (çökmelere) karşı sistem koruması
  - İzinsiz (veya kötü niyetli) erişime karşı güvenlik koruması
- Veri tabanı sisteminin **bakımı** (Maintain)
  - Gereksinimler zaman içinde değiştikçe sistemin gelişmesine izin vermesi.
- Verilerin **Sunumu** ve Görselleştirilmesi

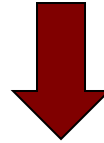




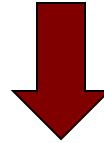
# Veri tabanı tasarlama aşamaları



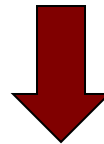
Gereksinimlerin özellikleri ve analizi (Dökümanlar)



Kavramsal tasarım (diyagramlar)



Mantıksal tasarım (normalizasyon)

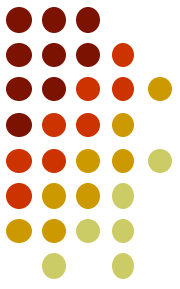


Fiziksel tasarım



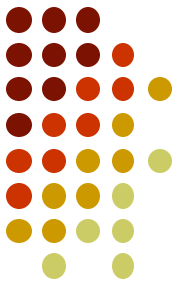
# Bir örnek

- ÜNİVERSİTE veritabanı
  - Bir üniversite ortamındaki öğrenciler, dersler ve notlarla ilgili bilgiler saklanmak isteniyor.
- Veri kayıtları / varlıklar
  - ÖĞRENCİ
  - DERS
  - GRUPNO (DERSİN)
  - NOT\_RAPORU
  - ÖN\_KOŞUL



## Bir örnek(devam)

- Her veri ögesi için veri türü belirleyin.
  - Alfabetik karakter dizisi
  - Tamsayı, vb.
- Kayıtlar arasındaki ilişkileri belirtin
  - BÖLÜMLER belirli DERSLER içindir.
  - ÖĞRENCİLER GRUP(Ders) a kayıtlıdır.
  - KURSLARın Önkoşul Dersleri vardır.



# Bir örnek(devam)

- ÜNİVERSİTE veri tabanını oluşturmak
  - Her öğrenci, ders, bölüm, not\_raporu ve önkoşulları uygun dosyada kayıt olarak göstermek için verileri depolayın.
- Veri Manipülasyonunu ihtiyaç halinde kullanın.
  - sorgulama (query)
    - seçme (select)
    - ekleme (insert)
    - güncelleme(update)
    - silme (delete)

## ÖĞRENCİ

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

## DERS

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

## GRUPNO

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

## NOT RAPORU

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

## ÖNKOŞUL

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310



# Geleneksel dosya işleme ve veritabanı yaklaşımı



- Geleneksel **dosya işleme**
  - Her kullanıcı, belirli bir yazılım uygulaması için gereken dosyaları tanımlar ve uygular.
- **Veri tabanı yaklaşımı**
  - Tek depo, bir defa tanımlanmış ve daha sonra çeşitli kullanıcılar tarafından erişilen verileri tutar.

# Veri Tabanı Yaklaşımının Temel Özellikleri



- Bir veritabanı sisteminin **kendi kendini tanımlayan yapısı**
  - Bir **VTYS kataloğu**, belirli bir veritabanının tanımını (örneğin, veri yapıları, türleri ve kısıtlamaları) saklar.
  - Bu tanıma **meta-veri** denir.
  - Bu, DBMS yazılımının farklı veritabanı uygulamalarıyla çalışmasına izin verir (örneğin, bir üniversite DB, bir bankacılık DB).
- Verilerin **çoklu görünümünün** desteklenmesi
  - Her kullanıcı, yalnızca o kullanıcının ilgilendiği verileri açıklayan veritabanının farklı bir görünümünü görebilir.



# Veri Tabanı Yaklaşımının Temel Özellikleri



- Programlar ve veriler arasında yalıtım ve veri soyutlama
  - Program-veri bağımsızlığı.
  - VTYSS erişim programlarını değiştirmek zorunda kalmadan veri yapılarının ve depolama organizasyonunun değiştirilmesine izin verir.
- **Veri soyutlama(Data abstraction):** Program veri bağımsızlığını ve program işlem bağımsızlığını sağlar.
  - Bir veri modeli (data model)(kavramsal gösterimi), depolama ayrıntılarını (veya işlemlerin uygulanmasını) gizlemek ve kullanıcılara veritabanının kavramsal bir görünümünü sunmak için kullanılır.
  - Programlar veri depolama ayrıntılarından ziyade veri modeli yapılarına atıfta bulunur.



## İLİŞKİLER

Relation_name	No_of_columns
STUDENT	4
COURSE	4
SECTION	5
GRADE_REPORT	3
PREREQUISITE	2

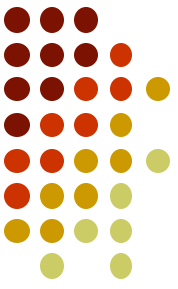
## KOLONLAR

Column_name	Data_type	Belongs_to_relation
Name	Character (30)	STUDENT
Student_number	Character (4)	STUDENT
Class	Integer (1)	STUDENT
Major	Major_type	STUDENT
Course_name	Character (10)	COURSE
Course_number	XXXXNNNN	COURSE
....	....	....
....	....	....
....	....	....
Prerequisite_number	XXXXNNNN	PREREQUISITE

# Veri Tabanı Yaklaşımının Temel Özellikleri

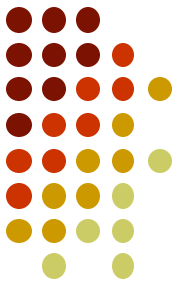


- **Veri paylaşımı ve çok kullanıcılı işlem işleme**
  - Bir dizi eşzamanlı kullanıcının veri tabanından aynı anda veri almalarını ve güncelleme yapmalarını sağlamak.
  - VTYS içindeki eşzamanlılık kontrol yazılımı aynı verileri güncellemeye çalışan birçok kullanıcının bunu kontrollü bir şekilde yapmasını ve böylece güncellemelerin sonucunun doğru olmasını sağlar. (VTYS, her işlemin doğru bir şekilde yürütülmesini veya iptal edilmesini garanti eder)
  - Kurtarma alt sistemi, her tamamlanan işlemin kalıcı olarak veritabanına kaydedilmesini sağlar.
  - OLTP (Online Transaction Processing) veritabanı uygulamalarının önemli bir parçasıdır. (örn. havayolu uçuş rezervasyonu uygulaması) Bu, saniyede yüzlerce eşzamanlı işlem gerçekleştirilmesine izin verir.



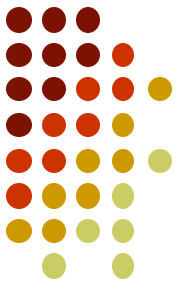
# Veri Tabanı Kullanıcıları

- Kullanıcılar
  - Veri tabanı içeriğini gerçekten kullanan ve kontrol edenler ile veri tabanı uygulamaları tasarlayan, geliştiren ve bakımını yapanlar (“**Sahnedeki Aktörler**” olarak adlandırılanlar) ve
  - VTYS yazılımı ve ilgili araçları tasarlayan ve geliştirenlerin yanı sıra bilgisayar sistemleri işletmecileri (“**Sahnenin Arkasındaki İşçiler**” olarak adlandırılır).



# Sahnedeki Aktörler

- **Veri tabanı yöneticileri (VTY)** şunlardan sorumludur:
  - Veritabanına erişim izni verme
  - Kullanımını koordine etmek, izlemek ve kontrol etmek
  - SW ve HW kaynaklarının edinilmesi (edinilmesi)
- **Sistem analistleri**
  - Son kullanıcıların gereksinimlerini belirleme



# Sahnedeki Aktörler(devam)

- **Veri tabanı tasarımcıları** şunlardan sorumludur:
  - Saklanacak verileri tanımlama
  - Bu verileri temsil etmek ve saklamak için uygun yapıları seçmek

(Son kullanıcıların ihtiyaçlarını anlamak için son kullanıcılarla (veya sistem analistleriyle) iletişim kurmaları gerekir)
- **Veritabanı Uygulayıcısı(DB Implementer):**
  - Söz konusu özelliklerin veri tabanı olarak uygulanması
- **Uygulama Programcısı(Application Programmer):**
  - Belirtilen özellikleri DB ile etkileşime giren programlar olarak uygular.



# Sahnedeki Aktörler(devam)

- **Son kullanıcılar**

- İşleri veritabanına erişim gerektiren kişiler

- **Türleri**

- Gündelik son kullanıcılar (Casual)
- Naif veya parametrik son kullanıcılar (Naïve (inexperienced) or Parametric)
- Gelişmiş son kullanıcılar (Sophisticated)
- Bağımsız kullanıcılar (Stand-alone)



# Sahnenin Arkasındaki İşçiler

- **VTYS tasarımcıları ve uygulayıcıları**
  - VTYS modüllerini ve arayüzlerini bir yazılım paketi olarak tasarlayıp ve uygularlar
- **Araç (Tool) geliştiricileri**
  - Araçlar tasarlayıp ve uygularlar
- **Operatörler ve bakım(maintenance) personeli**
  - Veritabanı sistemi için donanım ve yazılım ortamının çalıştırılmasından ve bakımından sorumludur.



# VTYS Yaklaşımını Kullanmanın Avantajları



- Verileri yalnızca bir yere yerleştirerek tekrarı denetleme
  - Veri normalizasyonu
  - denormalization
- Yetkisiz erişimi kısıtlama
  - Güvenlik ve yetkilendirme alt sistemi
  - Ayrıcalıklı (avantajlı) yazılım

(Bazı kullanıcıların yalnızca verileri almasına izin verilirken, diğerlerinin almasına ve güncellemesine izin verilir.)

# VTYS Yaklaşımını Kullanmanın Avantajları(devam)



- Program nesneleri için kalıcı depolama sağlama
  - C ++ içindeki karmaşık nesne, nesne yönelimli bir VTYS'de kalıcı olarak depolanabilir.
- Empedans uyumsuzluğu problemi
  - Nesneye yönelik veritabanı sistemleri genellikle veri yapısı uyumluluğu sunar

# VTYS Yaklaşımını Kullanmanın Avantajları(devam)



- Depolama yapıları (örneğin, genellikle ağaç veya karma veri yapısına dayanan endeksler) ve verimli sorgu işleme ve optimizasyon için arama teknikleri sağlama
- Belirli bir sorgu için gereken VT(DB) kayıtlarını işlemek için bu kayıtların diskten belleğe kopyalanması gerekir. Bu nedenle, DBMS genellikle veritabanının parçalarını ana bellek tamponlarında tutan bir tamponlama tamponu (veya önbellekleme-önbellek) modülüne sahiptir. Diğer durumlarda, VTYS disk verilerini tamponlamak için işletim sistemini kullanabilir.

# VTYS Yaklaşımını Kullanmanın Avantajları(devam)



- Yedekleme ve kurtarma sağlama
  - VTYS'nin yedekleme ve kurtarma alt sistemi kurtarma işleminden sorumludur (HW ve SW hatalarından kurtarma için)
- Farklı kullanıcı sınıflarına birden fazla kullanıcı arayüzü sağlama
  - Grafiksel kullanıcı arayüzleri (GUI'ler)
    - Hem form stili hem de menü odaklı arayüzler, grafiksel kullanıcı arayüzleri (GUI) (ör. Web GUI arayüzü) olarak bilinir. GUI belirtmek için birçok özel dil ve ortam vardır.
- Veriler arasındaki karmaşık ilişkileri göstermek
  - Birçok yönden birbiriyle ilişkili sayısız veri türü içerebilir.

# VTYS Yaklaşımını Kullanmanın Avantajları(devam)



- Bütünlük kısıtlarını zorlama
  - Referans bütünlüğü kısıtı
    - Her bölüm kaydı bir ders kaydı ile ilgili olmalıdır
  - Anahtar veya benzersizlik kısıtlaması
    - Her kurs kaydının, Kurs\_numarası için benzersiz bir değeri olması gerekir
  - İş kuralları
  - Veri modelinin içsel kuralları
  - Her veri ögesi için bir veri türü belirtme

# VTYS Yaklaşımını Kullanmanın Avantajları(devam)



- Tümdengelimli ve etkin kuralları kullanarak saklanan verilerden çıkarım ve işlem yapılmasına izin verilmesi
  - Tümdengelimli veritabanı sistemleri
    - Tümdengelim kurallarını tanımlamak için yetenekler sağlama
    - Depolanan veritabanı bilgilerinden yeni bilgilerin çıkarılması
- Tetikçiler (Trigger)
  - Tetikleyiciler tablolarla ilişkilendirilir. Tetikçi, tablodaki güncellemeler tarafından etkinleştirilen ve bazı diğer tablolara ek işlemler gerçekleştiren, mesaj gönderen vb.
- Saklı yordamlar (Stored procedures)
  - Kuralları uygulamak için daha çok ilgili prosedürlere popüler olarak saklı prosedürler denir; Genel VT(DB) tanımının bir parçası olurlar ve belirli koşullar yerine getirildiğinde uygun şekilde çalıştırılırlar.

# VTYS Yaklaşımını Kullanmanın Avantajları(devam)



- VT(DB) yaklaşımını kullanmanın diğer etkileri
  - Standartları uygulama potansiyeli:
    - Standartlar, veri ögesi adlarını, ekran biçimlerini, ekranları, rapor yapılarını, meta-verileri (verilerin tanımı), Web sayfası düzenlerini vb. gösterir.
  - Azaltılmış uygulama geliştirme süresi:
    - Her yeni uygulamanın eklenmesi için artan zaman azalır.
  - Veri yapılarını değiştirme esnekliği:
    - Yeni gereksinimler tanımlandıkça veritabanı yapısı gelişebilir.
  - Güncel (güncel) bilgilerin kullanılabilirliği:
    - Havayolu, otel, araç rezervasyonları gibi çevrimiçi işlem sistemleri için son derece önemlidir.
  - Ölçek ekonomileri:
    - Verileri ve uygulamaları departmanlar arasında birleştirerek (birleştirerek) israfli kaynak ve personel örtüşmesinden kaçınılabilir.

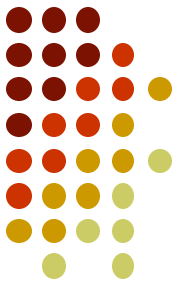
# Veri Tabanı Uygulamalarının Kısa Tarihçesi



- İlk Veritabanı Uygulamaları:
  - Hiyerarşik ve Ağ Modelleri 1960'lı yılların ortalarında tanıtıldı ve yetmişli yıllarda egemen oldu.
  - Bu modeller benzer yapıdaki çok sayıda kaydı içerir.
  - Dünya genelinde veri tabanı işlemenin büyük bir kısmı hala bu modellerin, özellikle de hiyerarşik modelin kullanılmasıyla gerçekleşiyor.
- İlişkisel Model Tabanlı Sistemler:
  - İlişkisel model ilk olarak 1970 yılında tanıtıldı, IBM Research ve birkaç üniversitede yoğun olarak araştırıldı ve denendi.
  - İlişkisel DBMS Ürünleri 1980'lerin başında ortaya çıktı.
  - Veri soyutlama ve uygulama esnekliği sağlamak için verilerin fiziksel depolanmasını kavramsal temsilinden ayırır
  - Veri sunumu ve sorgulaması için matematiksel bir temel sağlar.

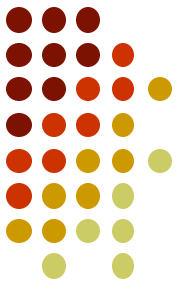


# Veri Tabanı Uygulamalarının Kısa Tarihçesi(devam)



- Nesneye yönelik ve ortaya çıkan uygulamalar:
  - Nesne Yönelimli Veri Tabanı Yönetim Sistemleri (OODBMS'ler) 1980'lerin sonunda ve 1990'ların başında, CAD gibi ve diğer özel uygulamalarda karmaşık veri işleme ihtiyacını karşılamak (sağlamak) için tanıtıldı: mühendislik tasarımı, multimedya yayıncılık ve üretim sistemleri
    - Kullanımları fazla sürmedi.
  - Pek çok ilişkisel DBMS, nesne veritabanı kavramları dahil etmiş ve nesne-ilişkisel DBMS'ler (ORDBMS'ler) adı verilen yeni bir kategoriye öncülük etmiştir.
  - Genişletilmiş ilişkisel sistemler başka özellikler de ekler (örneğin, multimedya verileri, XML ve diğer veri türleri için)

# Veri Tabanı Uygulamalarının Kısa Tarihçesi(devam)

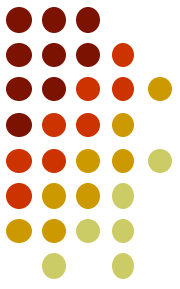


- Web ve E-Ticaret Uygulamalarındaki Veriler:
  - Web, sayfalar arasında bağlantı içeren HTML'deki (Köprü metni biçimlendirme dili) verileri içerir.
  - Bu yeni bir dizi uygulamaya yol açmıştır ve E-ticaret XML (Yeni İşaretleme Dili) gibi yeni standartlar kullanmaktadır.
    - Çeşitli veritabanı türleri ve Web sayfaları arasında veri alışverişi için genişletilmiş işaretleme dili (XML) birincil standardı (örneğin, XML kullanan Web'deki e-ticaret uygulamaları)
  - PHP ve JavaScript gibi script programlama dilleri, kısmen bir veritabanından oluşturulan dinamik Web sayfalarının oluşturulmasını sağlar.
    - Ayrıca Web sayfalarında veritabanı güncellemelerine izin verir

# Veri Tabanı Uygulamalarının Kısa Tarihçesi(devam)



- Yeni uygulamalar için veritabanı yeteneklerini genişletme
  - Uygulamalar için özel gereksinimleri daha iyi desteklemek için uzantılar
  - Kurumsal Kaynak Planlaması (ERP)
  - Müşteri ilişkileri yönetimi (CRM)
- Veri tabanı (DB) ve veri alımı (information retrieval)
  - Bilgi alma (IR) (information retrieval)
    - Kitap, el yazması ve çeşitli kütüphane temelli makale türleri ile ilgilenme



# Veritabanı Yeteneklerini Geniřletmek

- VTYS'lere ařađıdaki alanlarda yeni iřlevler eklenmektedir:
  - Bilimsel Uygulamalar
  - XML (uzatılabilir İřaretleme Dili)
  - Grnt Depolama ve Ynetimi
  - Ses ve Video Veri Ynetimi
  - Veri Depolama ve Veri Madenciliđi
  - Mekansal Veri Ynetimi
  - Zaman Serileri ve Tarihsel Veri Ynetimi
- Yukarıdakiler, yeni veri tipleri, karmařık veri yapıları, yeni iřlemler ve veri tabanı sistemlerinde depolama ve indeksleme řemalarının birleřtirilmesinde yeni arařtırma ve geliřtirmelere yol amaktadır.



# Ne Zaman VTYS Kullanılmamalı

- DBMS kullanmanın ana inhibitörleri (maliyetleri):
  - Yüksek ilk yatırım ve ek donanım için olası bir ihtiyaç.
  - Genellik, güvenlik, eşzamanlılık kontrolü, kurtarma ve bütünlük işlevleri sağlamak için ek yük.
- Bir DBMS gerekli olmadığında (normal dosyaları kullanmak daha uygun olabilir):
  - Basit, iyi tanımlanmış veritabanı uygulamalarının hiç değişmesi beklenmiyorsa
  - Verilere çok kullanıcı ile erişim yoksa
  - Sınırlı depolama kapasitesine sahip gömülü sistemler varsa
  - VTYS ek yükü nedeniyle karşılanamayan katı, gerçek zamanlı gereksinimler varsa
- Hiçbir VTYS ne gerek olmadığında :
  - Veri tabanı sistemi, modelleme sınırlamaları nedeniyle verilerin karmaşıklığını kaldıramazsa
  - Veritabanı kullanıcılarının DBMS tarafından desteklenmeyen özel işlemlere ihtiyacı varsa.