

BİLGİ SİSTEMLERİ

①

SLAYT 1 ⇒ Giriş

Bilgi = Öğrenme, araştırma veya çözüm yolu ile elde edilen gerçek, malumat, vukuf.

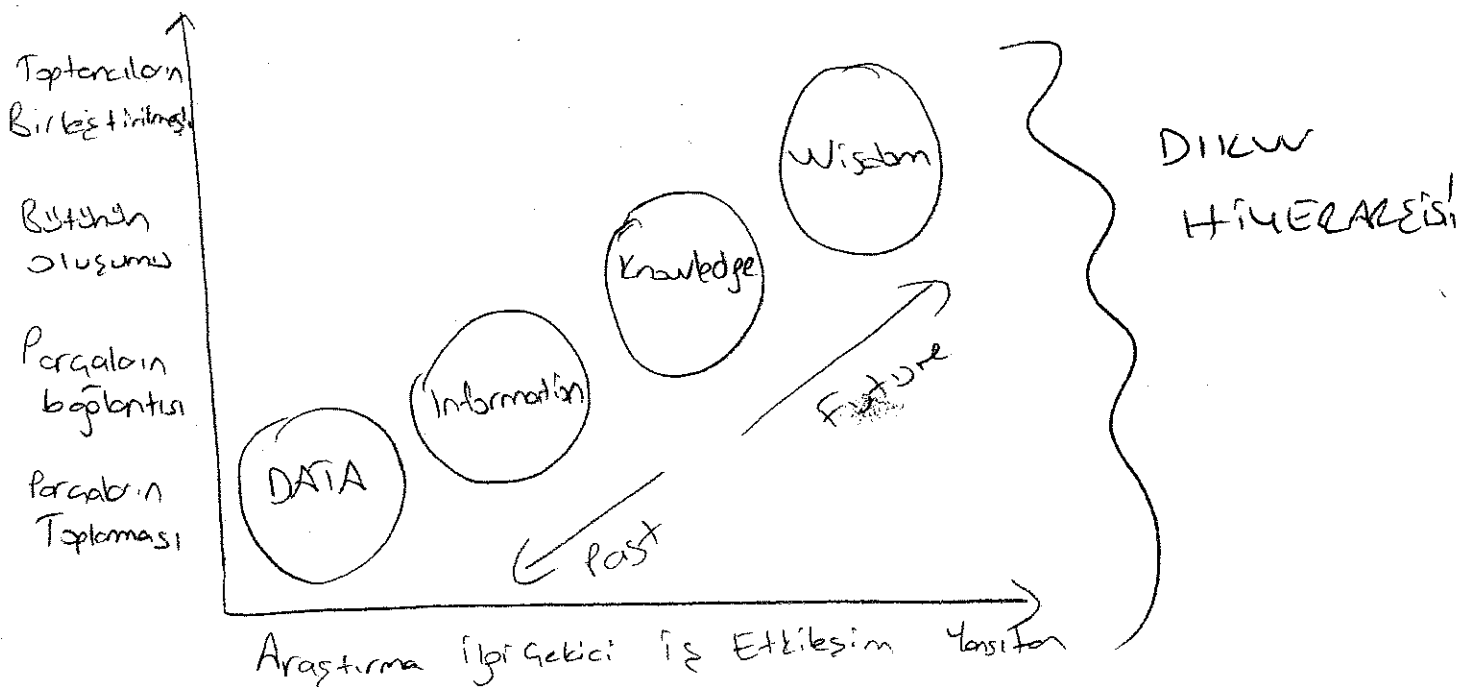
İrfan = Bilmek, anlayış, tecrübe ve zekadan ileri gelen zihni kemal.

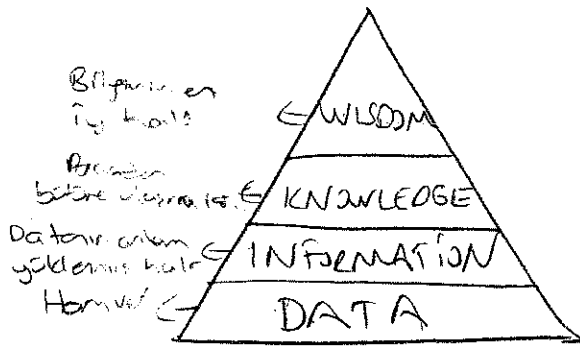
Marifet = \*Bilme, bir şeyi cüz'î ve ihtik bilmek.  
\* İlim ve fenlerle tahsil olunan malumat.

DİKK HİYERARŞİSİ

\* Söz konusu yapısal yapıyı temsil eden bir model sınıfına geçerek bir şekilde atıfta bulunur.

\* Tipik olarak bilgi, bilgi açısından bilgi, bilgi açısından bilgelik ve bilgi bakımından bilgelik olarak tanımlanır.





⇒ Knowledge Pyramid  
"Bilgi Piriimid"

## SLAYT 2 ⇒ BİLGİ SİSTEMLERİ

**Bilgi Sistemlerinin Tanımı** = Bilgi sistemleri toplumlarda ve organizasyonlarda bilginin toplanması, işlenmesi, depolanması, dağıtılması ve kullanılması ile ilgili stratejik yönetsel ve operasyonel işlemlerde işlenen profesyonel ve akademik bir disiplindir.

\* Tipik olarak bir bilgi sistemi dijital bilgiyi toplama ve analiz etmekte kullanılan insanlar, prosedürler, veriler, yazılım ve donanımları içerir.

## Bilgi Sistemlerinin Türleri

- \* Hareket İşleme (İşlem İşleme) Sistemleri
- \* Yönetim Bilgi Sistemleri
- \* Karar Destek Sistemleri
- \* Yönetici Bilgi Sistemleri

## Bilgi Sistemlerinin Bileşenleri

- \* İnsanlar
- \* Ekipmanlar
- \* Prosedürler
- \* Veri

## Hareket Yönetim Sistemleri (HYS)

(2)

\* Hareket yönetim sistemleri bir organizasyonun günlük hareketlerini toplayan, depolayan, değişiklik yapan ve erişime açan bilgi sistemleridir.

\* HYS'ler işletiminde genellikle paran hareketini ve bir şirketin günlük işlerini güvenli bir şekilde kayıt altına alırlar.

## İşlem İşleme Sistemi (TPS)

\* Bir bilgi sistemi türüdür.

\* TPS'ler bir kuruluşun işlemlerini toplar, saklar, değiştirir ve geri alır.

\* Bir işlem sonunda bir bilgi sisteminde saklanan verileri üretir veya düzenleyen bir objedir.

ACID: Bilgisayar bilimlerinde ACID, veritabanı işlemlerinin güvenilir bir şekilde işlenmesini garanti eden bir dizi özelliktir.

## Atomicity

\* Veritabanı modifikasyonlarının bir "tümünü veya hiçbir" kuralını izlenmesini gerektirir. İşlemin bir kısmı başarısız olursa tüm işlem başarısız olur ve veritabanı durumu değişmeden kalır.

## Tutarlılık Özelliği

\* Veritabanının gerçekleştirdiği herhangi bir işlemin onu tutarlı bir durumdan diğere götürmesini sağlar.

\* Tutarlılık veritabanına sadece tutarlı verilerin yükleneceğini belirtir.

## İsyan

İsyan hiçbir işlemin başka bir işleme müdahale etmemesi

ni ifade eder.

• aynı satırların aynı anda çalışmasını etkilemesi mümkün değildir.

## Doyanıklılık

\* Doyanıklık bir işlem gerçekleştirildikten sonra, öyle kabaca onomina gelir.

Özellikler bir toadhet edilen işlem elektrik kesintisi / arızasına hatasız karşı konur ve sistem tarafından kaybedilemez ve böylece tamamlanacağı garanti edilebilir.

## HYS ÖZELLİKLERİ

- \* Hızlı cevap
- \* Güvenilirlik
- \* Esnek olma
- \* Kontrolü işlemler

Toplu İşlemler = Bilgisayarın ve işlemci gücünün düşük olduğu zamanlarda toplu işlem sistemleri geliştirilmiştir.

\* Toplu işlemlerde, işlenmesi gereken tüm bilgiler önce toplanır, sonra toplu bir halde işlenir.

## Gerçek Zamanlı İşleme (Real Time Processing)

\* Veri işleme aynı anda aynı yerde gerçekleşir.

Ör: Pos makineleri, ATM'ler

Burada iki önemli kavram gözetilmelidir.

- Concurrency (Aynı Anda Erişim)

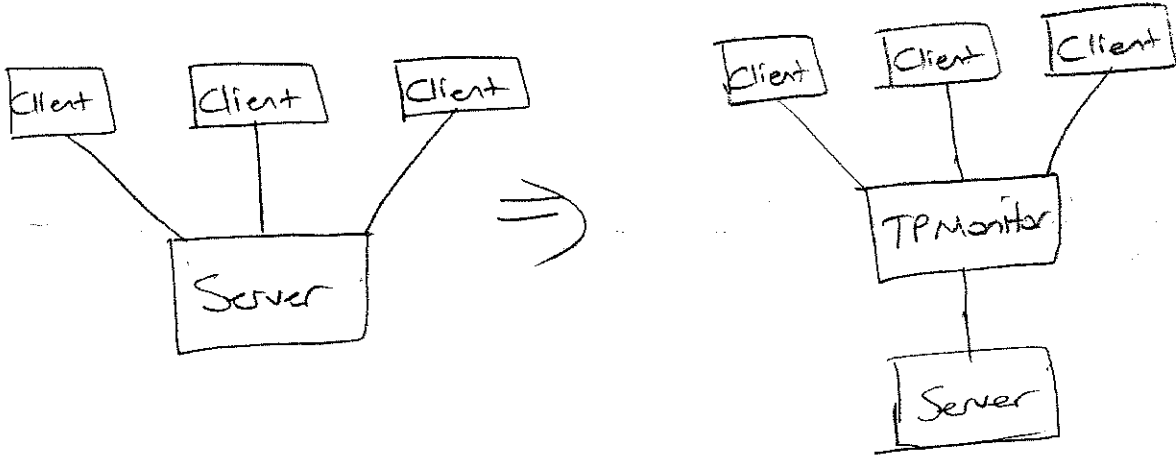
\* Aynı veriyi aynı anda birden fazla kullanıcı değiştirememelidir.

- Atomicity

\* İşlemin bütün adımları tamamlanmalı veya işlem gerçekleşmemiş kabul edilmeli ve tüm değişiklikler geri alınmalıdır.

## Hareket İzleme Monitörü

(3)



## SLAYT 3 ⇒ VERİNİN TEMSİLİ

### Veri - Kitap Listesi

Anna Karenina	Lev N. Tolstoy	Sosyal	2002	975 738425019.9
Güller Adı	Umberto Eco	Can	2009	975 510245032
Semerkent	Amin Maabul	Yapı Kredi	2010	975 081003916
Leyla ile Mecnun	İstikbal Pala	Kepi	2010	975 89503868
Akşam Güneşi	Reşat Nuri Günekkı	İnkılap	2010	975 102660621

\* Dosyalarda saklanabilir.

\* Sorgulama imkanı yok.

\* Özel programlarla arama ve sorgulama yapılabilir.

### Veritabanı

\* Veritabanı ile yukarıda yazılan tabloya "İsim, Yazar, Yayıncı, Yayıncılık, ISBN, Fiyat" gibi sütun türleri belirlenebilir.

\* Yeni veri ekleme oldukça kolaydır. ö.e; Fiyat türü  
sorgulama yapılabilir.

## Geri Dönüşlülük

- \* Veritabanına yeni sütun eklenmesi
- \* Veriye daha önce düşünülmemiş bir görüntünün eklenmesi daha önce geliştirilmiş temsil yöntemlerini barındırır.
- \* Verinin yapısının her zaman önceden bilinmesi ve bunlara göre veri analizi programının yazılması mümkün değildir.

## Yeni bir veri modeli ihtiyacı

- \* Gerçek ve zengin bir yapı
- \* Gelişmeye açık, bilinmeyen özelliklere kolay adapte olabilir
- \* Düzensiz yapıları destekleyebilecek
- \* Yapılandırılmış fakat heterojen veri kaynaklarını birleştirebilmeli
- \* Tag ve linklerden oluşan text türü veri
- \* Farklı veri modellerinin kombinasyonu (ilişkisel, hiyerarşik, text)

## Markup Languages

- \* 1970'de Charles Goldfarb, Ed Mosher ve Ray Lorie IBM'de çalışırken GML'i buldular.
- \* GML yapısal etiketler taşıyan teknik dokümanları temsil etmek için, bir yöntemi tarif etmek için geliştirilmiştir.
- \* Bulbun isminin kısaltması (G-M-L)
- \* Goldfarb 'mak-up language' terimini icat etti ve daha sonra bu isim SGML olarak değiştirildi.
- \* ISO 1986'da SGML'i bir standart olarak kabul etti.

HTML

- \* SGML bir dizi işaretleme dili geliştirmek için kullanıldı.
- \* 1991'de SGML kullanılarak HTML geliştirildi.
- \* SGML çok efektif fakat kompleks, HTML ise çok basit fakat belirli bir dizi tag ile kısıtlıdır.
- \* HTML verinin nasıl gösterileceği ile ilgilenir.

Örnek bir HTML kodu

```

<html>
<body>
<table>
  <tr>
    <td> İsim </td>
    <td> Yaş </td>
  </tr>
  <tr>
    <td> Anna Karenina </td>
    <td> Lev. N. Tolstoy </td>
  </tr>
  <tr>
    <td> Gülün Adı </td>
    <td> Umberto Eco </td>
  </tr>
</table>
</body>
</html>

```

SGML  
Çok efektif  
Fakat kompleks

HTML  
Çok basit  
Fakat belirli  
bir dizi tag  
ile sınırlı

## XML

- \* SGML kadar etkili; fakat o kadar karmaşık olmayan
- \* HTML kadar basit fakat kısıtlı olmayan bir markup diline olan ihtiyaç
- \* XML bu boşluğu doldurmak için geliştirildi.

## XML vs HTML

- \* XML verinin içeriğini tarif eder.
- \* HTML verinin nasıl gösterileceğini anlatır.
- \* XML verinin aktarımı ve depolanması amacıyla geliştirilmiştir.
- \* XML tek başına hiçbir iş yapmaz.

### XML

- \* Verinin içeriğini tarif eder
- \* Verinin aktarımı ve depolanması amacıyla geliştirilmiştir
- \* Tek başına hiçbir iş yapmaz

### HTML

- \* Verinin nasıl gösterileceğini



## SLAMT 4 ⇒ XML

(5)

\* Yapısal veriler taşıyan dökümanları biçimlemek için geliştirilmiş bir dil.

\* İnternette veri değişiminde ilgili eksikliği kapatmak amaçlanmıştır

XML	HTML
⇒ İhtiyaç göre oluşturulabilir taglar	⇒ Belli sayıda tag
⇒ Farklı yönelimli	⇒ Verinin gösterimini esas alır.
⇒ Standart veri değişim biçimi	⇒ Veri doğrulama yeteneği yok.
⇒ Farklı çıktı biçimlerine izin verir.	⇒ Tek tür, standart çıktı

## XML NE İÇİN KULLANILIR ?

\* İnternette verilerin bir yerden başka bir yere aktarılması için

\* Özel olan uygulamaları için geliştirilebilir.

## XML'in Avantajları

\* Veriyi HTML'den ayırır.

\* Veri paylaşımını basitleştirir.

\* Veri değişimini basitleştirir.

\* Platform bağımsızlığını kolaylaştırır.

\* Verinin daha fazla tüketiciye açık olmasını sağlar.

## ÖRNEK:

<Kitap>

<İsim>Anna Karenina</İsim>

<Yazar>Lev Tolstoy</Yazar>

<Yayınevi>Can</Yayınevi>

<Yayınyılı>2008</Yayınyılı>

<ISBN>12345678910</ISBN>

<Fiyat>38,5</Fiyat>

</Kitap>

## XML KURALLARI

- \* Bir XML dokümanı bir veya birden fazla element içerir.
- \* Her elementin bir ismi vardır. Başka elementler barındırabilir ve nitelikler taşıyabilir.
- \* Etiketler başlama etiketi ve bitiş etiketi olmak üzere çift olarak yazılır.
- \* Etiketler doğru yazılmalıdır.

⇒ `<Kitap><İsim></Kitap></İsim>` ⇒ Hatalı Yazım

⇒ `<Kitap><İsim></İsim></Kitap>` ⇒ Doğru Yazım

- \* Boş etiketler kısaltılabilir.

⇒ `<Kitap></Kitap>` şeklinde bir boş etiket  
`<Kitap/>` şeklinde kısaltılabilir.

- \* Büyük, küçük harflere duyarlıdır.

- \* Nitelikler eşit işaretlik ayrılmış isim-değer çiftleridir.

⇒ `<Öğrenci No="12345">Ali Demir</Öğrenci>`

XML'de referans vermek

`<Soygöçü>`

`<Kişi id="1">`

`<İsim>Ali</İsim>`

`</Kişi>`

`<Kişi id="2">`

`<İsim>veli</İsim>`

`</Kişi>`

`<Kişi id="3">`

`<İsim>Ayşe</İsim>`

`<Çocuklar ids="1 2"></Çocuklar>`

`</Kişi>`

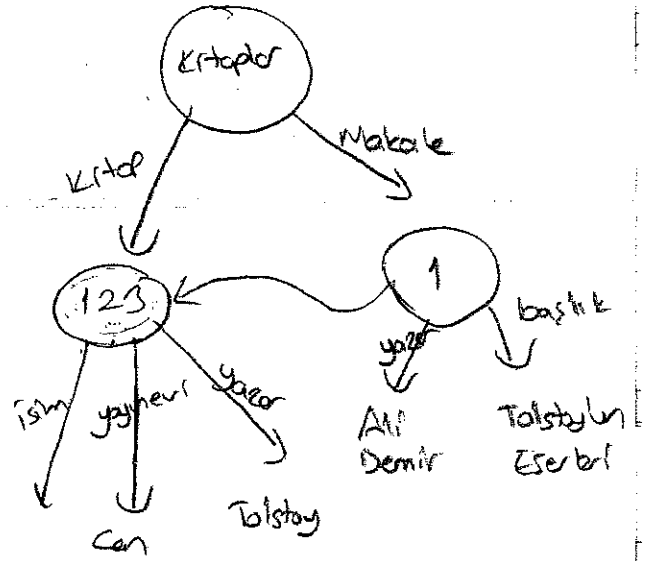
`</Soygöçü>`

# XML Veri Modeli: Ağaçlar

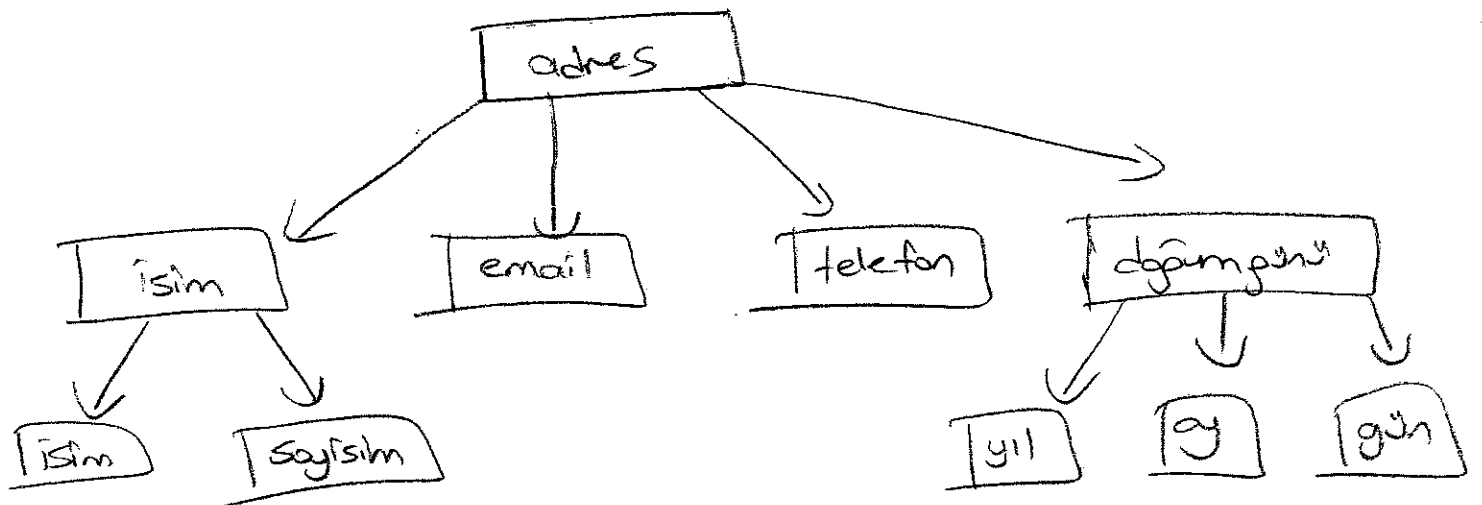
```

<Kitaplar>
  <Kitap id="123">
    <isim>Anna Karenina</isim>
    <Yazar>Lev Tolstoy</Yazar>
    <Yayinevi>Can</Yayinevi>
    <Yayinyili>2008</Yayinyili>
    <ISBN>123456789</ISBN>
    <Fiyat>35</Fiyat>
  </Kitap>
  <Makale ref="123" id="1">
    <Yazar>
      <isim>Ali Demir</isim>
    </Yazar>
    <Başlık>Tolstoy'un Eserleri</Başlık>
  </Makale>
</Kitaplar>

```



## XML AĞAÇLARI



## İlişkisel Veritabanı - XML

İsim	telefon
ali	123
ayşe	456
fatma	789

<Kişiler>

<Kişi>

<İsim> ali </İsim>

<telefon> 123 </telefon>

</Kişi>

<Kişi>

<İsim> ayşe </İsim>

<telefon> 456 </telefon>

</Kişi>

<Kişi>

<İsim> fatma </İsim>

<telefon> 789 </telefon>

</Kişi>

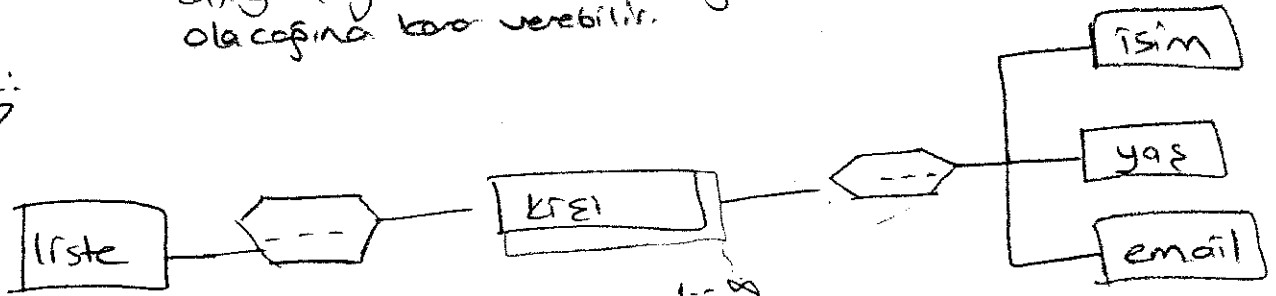
</Kişiler>

DTD  $\Rightarrow$  XML dilinin bir parçasıdır ve yazıldığı XML dokümanları için gramer olarak işlev görür.

XML Schema  $\Rightarrow$  XML Schema aynı zamanda XML dokümanıdır.  
• Elementer simple ve complex türler olarak 2'ye ayrılmıştır.

• Ağacı yapısının nasıl olacağı ve bir elementin kaç çocuğu olacağına karar verebilir.

Öz:  
 $\rightarrow$



## XSLT

\* XSLT bir XML dökümanını başka birisine dönüştürmek için kullanılır. (Genelde HTML)

\* XSLT dönüştürme programları bir XML dosyasını girer olarak alıp başka bir XML dosyasını çıktı olarak verir.

\* Eğer çıktı HTML ise web tarayıcıları ile görüntülenebilir.

\* XML verisini kolaylıkla göstermek için güzel bir yol.



## SLAYT 5 ⇒ DOM

\* DOM bir W3C standardıdır.

\* XML ve HTML gibi dokümanlara erişim için standart bir yol tanımlar.

⇒ DOM standart modeller tanımlayan 3 farklı parça/seviye içerir.

→ Core DOM: Herhangi bir yapısal doküman için

→ XML DOM: XML dokümanları için

→ HTML DOM: HTML dokümanları için

XML DOM:

Bütün XML elementleri için nesne ve özellikleri ve bunlara erişim için gerekli metodları tanımlar.

⇒ Diğer bir deyişle DOM, XML elementlerine erişim, değiştirme, ekleme, silme gibi işlemler için bir standart sağlar.

DOM NODE:

\* DOM açısından XML içindeki herşey bir düğümdür. (node)

\* Dokümanın kendisi doküman düğümüdür.

\* XML elementlerinin içindeki yazılar yazı düğümüdür. (text node)

\* Herbir nitelik (attribute) nitelik düğümüdür.

\* Yorum düğümlerinde bulunabilir.

DOM AĞACI:

\* Ağacın başındaki herbir düğümün diğerleriyle hiyerarşik bir ilişkisi vardır.

\* Bu ilişkileri anlatmak için ebeveyn, çocuk ve kardeş terimleri kullanılır.

\* Ebeveyn düğümün çocukları vardır.

\* Aynı seviyedeki çocuklar kardeş olarak adlandırılırlar.

## Düğüm Ağacında İlişkiler:

- \* Bir düğüm ağacında en tepedeki düğüme kök denir.
- \* Kök düğüm hariç her düğümün sadece bir ebeveyni vardır.
- \* Bir düğümün sınırsız sayıda çocuğu olabilir.
- \* Çocuğu olmayan düğümlere yaprak denir.
- \* Aynı ebeveyni sahip aynı seviyedeki düğümlere kardeş denir.

## XML DOM PARSER:

- \* XML dokümanlarını ayrıştırmak için kullanılan nesneler uygulaması (parser) yazılmıştır.
- \* XML Dom modeli elemanlara erişim, değiştirmek, silmek, veya eklemek için metodlar içerir. Ancak öncelikle dokümanın Dom nesnesine yüklenmesi gerekir.

## Dom Özellikleri:

- ⇒ `x.nodeName` ⇒ x düğümünün ismi
- ⇒ `x.nodeValue` ⇒ x düğümünün değeri
- ⇒ `x.parentNode` ⇒ x düğümünün ebeveyn düğümü
- ⇒ `x.childNodes` ⇒ x düğümünün çocuğu olan düğümler
- ⇒ `x.attributes` ⇒ x düğümünün nitelikleri

## - SAX -

- \* SAX, DOM'a alternatif olarak geliştirilmiş bir sıralı erişim ayrıştırıcıdır.
- \* XML dokümanlarını okumak için gerekli araçları sağlar.
- \* SAX, obje-güdümlü olarak çalışır; akışa göre çalışan bir ayrıştırıcıdır.
- \* SAX ayrıştırıcısı karşılaştığı düğümleri sırasıyla işler ve bu işleme dokümanın sonuna kadar devam eder.

## SAX vs DOM

- \* Sax Dom'a göre daha küçük hafızaya ihtiyaç duyar.
- \* Dom; XML ağacının tamamen hafızaya yüklenmesine ihtiyaç duyar.
- \* Obj güdümlü yaklaşım nedeniyle SAX daha hızlıdır.
- \* XSLT ve XPATH gibi bazı teknolojiler XML ağacının tamamına ihtiyaç duyduğu için SAX kullanılmaz.



## SLAHT 6 ⇒ XPATH

- \* XPATH bir W3C standardidir.
- \* XPATH XML dökümanlarının kısımlarını seçmek için kullanılan bir sintakstir.
- \* XPATH sorguları XML olarak yazılmazlar. Ama genelde XSLT ile görünür bir şekilde kullanılır.
- \* Küçük bir programlama dilir gibi fonksyonları, testleri ve ifadeleri vardır.

### XPATH SYNTAX:

- \* XPATH'in eleman adresi tarif yaklaşımı işletim sistemlerindeki dosya adresleme yöntemlerine benzer.
- \* XPATH ifadesi işletildiğinde tipi düğüm seti, boolean, rakam veya string tipinde bir nesne döndürür.

### PARANTEZLER VE LAST() :

- \* Parantez içindeki bir rakam o sayıya denk gelen belirli bir elementi seçer.

Öz: /kütüphane/kitap[1]

↑  
Kütüphanedeki ilk kitabı seçer.

- \* Last() fonksyonu parantez içinde kullanılıncsa son sonucu seçer.

Öz: /kütüphane/kitap/bölüm[last()]

### YILDIZLAR :

Yıldız veya asterisk "wild card" olarak kullanılır. ve bu seviyedeki tüm elementler anlamına gelir.

Öz: /kütüphane/kitap/bölüm/\* ifadesi kütüphane içindeki her bir kitap elementinin her bölüm elementinin tüm çocuklarını seçer.

## Öznitelikler (Attributes):

\* Niteliklerin kendilerini veya belirli nitelik barındıran elementleri seçilebilirliğidir.

Öz: isimdeğer çifti,  $\langle \text{bölüm no} = "5" \rangle \Rightarrow$  bölüm no'su 5 olanlar

Öz: nitelik seçmek için,  $@no \Rightarrow$  no'su olan bütün nitelikler.

Öz:  $\text{bölüm}[@no] \Rightarrow$  no adında niteliği bulunan bölümler

Öz:  $\text{bölüm}[\text{not}@no] \Rightarrow$  no adında niteliği bulunmayanlar

Öz:  $\text{bölüm}[@*] \Rightarrow$  içinde attributes bulunan tüm bölümler

Öz:  $\text{bölüm}[@no='3'] \Rightarrow$  no adında bir niteliği olup değeri 3 olan tüm bölümler.

## Web Server

\* İnternet başlı olarak web sayfalarına veya diğer verilere erişebilmek için kullanılır.

\* En yaygın kullanımı web sitelerine evsahipliği yapmak.

\* Veri depolama ve kurumsal uygulamaları birleştirme gibi kullanım alanları da vardır.

## XML WEB SERVİS STANDARTLARI:

SOAP: Web servislerini işleyen XML tabanlı bir protokoldür.

Soap kullanarak

↳ Doküman Akılları

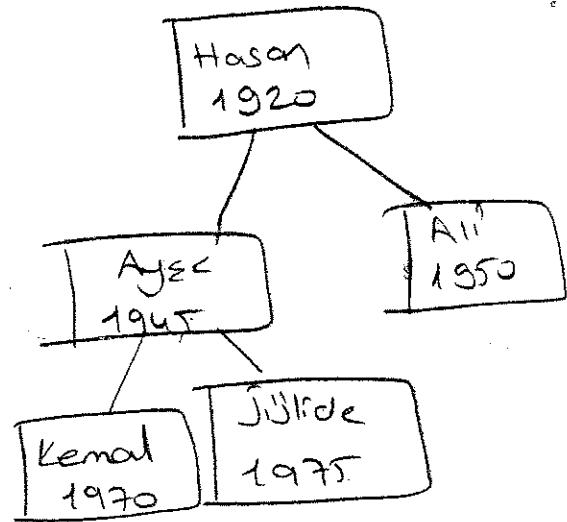
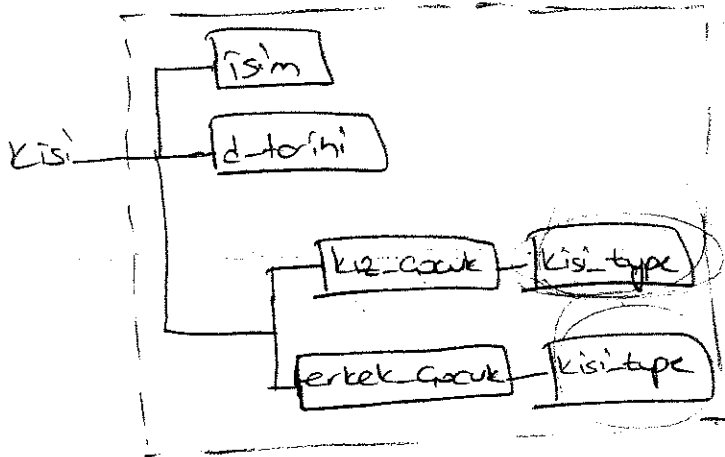
↳ Uzaktan metod çağırma yapılır.

WSDL: Web Servisleri için model sağlayan, web servisleri ve onlara nasıl erişileceğini tanımlayan XML tabanlı bir dildir.

↳ WSDL ile web servislerinin metodları ve adresleri belirtilir.

DDO1 →

## XML ÖRNEK



< Kisi >

< isim > Hasan < / isim >

< d\_tarihi > 1920 < / d\_tarihi >

< kiz-cocuk >

< isim > Ayşe < / isim >

< d\_tarihi > 1945 < / d\_tarihi >

< / kiz-cocuk >

< isim > Jülide < / isim >

< d\_tarihi > 1975 < / d\_tarihi >

< / kiz-cocuk >

< erkek-cocuk >

< isim > Kemal < / isim >

< d\_tarihi > 1970 < / d\_tarihi >

< / erkek-cocuk >

< / kiz-cocuk >

< erkek-cocuk >

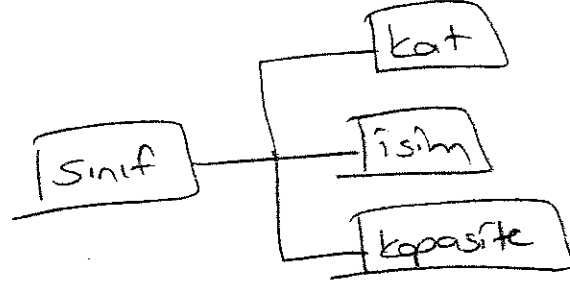
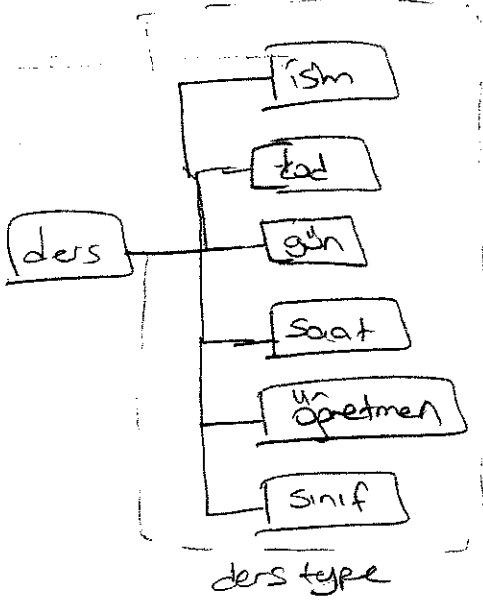
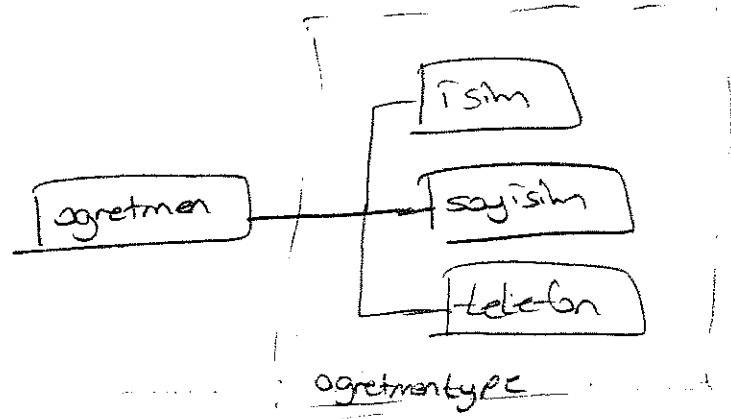
< isim > Ali < / isim >

< d\_tarihi > 1950 < / d\_tarihi >

< / erkek-cocuk >

< / Kisi >

## XML ÖRNEK:



<program>

<ders derstype="1">

<isim>matematik</isim>

<kod>123</kod>

<gun>arsamba</gun>

<saat>13:00</saat>

<ogretmen ogretmentype="1">

<isim>Meltem</isim>

<sayisim>Polat</sayisim>

<telefon>1234</telefon>

</ogretmen>

<sinif>

<kat>1</kat>

<isim>bd1</isim>

<kapasite>30</kapasite>

</sinif>

</ders>

</program>



## Karar Destek Sistemleri

- ↳ Karar verme sürecini hızlandıran ve kolaylaştıran sistemdir.
- ↳ Yöneticiye destek sağlar ona asla yöneticinin yerini almaz.

$$\text{İnsan Zekası} + \text{Bilgi Teknolojisi} + \text{Yazılım} = \text{KDS}$$

### 1.) Veri Odaklı Karar Destek Sistemleri

Büyük veri havuzlarını analiz eden sistemler

### 2.) Model Odaklı Karar Destek Sistemleri

"Eğer... ise" analizlerinin yapılması için bazı modeller kullanılarak tek başına bir sistemdir.

## Uzman Sistem

- ↳ Belirli bir alanın içinde bulunan uzmanlarından derlenmiş bilgilerle dayanarak düzenlenmiş gerekseyizdeki "eğer... ise...dır" biçimindeki kuralardan oluşur.

### Gesitleri:

- ↳ Bulanık Uzman Sistemler: Veriler üzerinde akıl yürütme yapan ve kesin bir sonuç vermayan bilgi programlarıdır.

- ↳ Yapay Sinir Ağları:

- ↳ Genetik Algoritmalar:

## BİLEŞENLERİ

- ↳ Kullanıcı Ara Birimi

- ↳ Bilgi Kazanım Modülü

- ↳ Bilgi Tabanı

- ↳ Karar Verme Mekanizması

