# IV1023 vt2022
# Avancerad Datahantering med XML

# Semistrukturerade data och XML

**nikos dimitrakas**
**nikosd@kth.se**
**08-161295**
**Rum 2423**

Läsanvisningar
Utdrag från Data on the Web
Kapitel 1, 4, 5, 6, 10 (speciellt 10.6) i kursboken
Delar av kapitel 31 i Database Systems (Connolly, Begg) upplaga 5





## Data - Metadata

- **Data**
  - **Johnny, Pasta, Lund, 2001-02-12, true, 677**
- **Metadata**
  - **namn, namn, stad, startdatum, skickad, vikt**
- **Typer av metadata**
  - **Struktur**
  - **Semantik**
  - **Katalog (klassificering)**
  - **Integration (mappning)**

# Struktur

- **Modellering**
  - TechTarget: Data modeling is the analysis of data objects that are used in a business or other context and the identification of the relationships among these data objects.
- **Databaslösningar**
  - **Relationsmodellen**
    - » **Tabeller, kolumner, domäner, nycklar, integritetsregler**
  - **Objekt-orientering/Objektdatabaser**
    - » **Klasser, attribut, referenser, regler**
  - **XML**
    - » **Element, attribut, regler**
  - **Annat**
    - » **?**





# Semantik

- **Betydelsen av data och metadata**
  - **Metadata**
    - » **namn**
    - » **pris**
    - » **vikt**
    - » **skickad**
  - **Semantik**
    - » **Det som unikt identifierar varje produkttyp vi har produkter av**
    - » **Antalet SEK som kunden måste betala inkl moms för ett exemplar**
    - » **Anger produktens vikt inkl förpackningen för ett exemplar i gram**
    - » **Sant om beställningen har lämnat vårt lager, annars falskt**

# Semistrukturerade data

- Ingen struktur (schemaless)
- Implicit struktur (self-describing)
  - metadata inbyggda i data
    - » inga data → inga metadata
- SSD

```
{namn:{för:"Kalle", efter:"Lind"},
 epost:"kalle@lind.nu",
 mobil:"07012345678"}

{namn:"Lisa",
 telefon:"0709999999"}
```



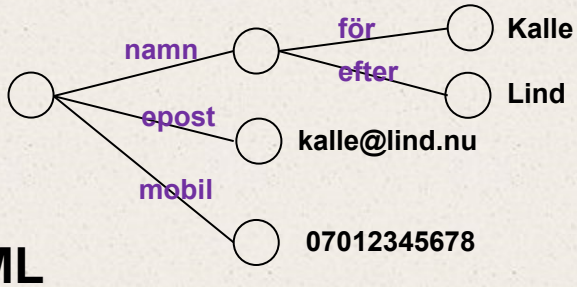


# Representationer

- SSD

```
{namn:{för:"Kalle", efter:"Lind"},
 epost:"kalle@lind.nu",
 mobil:"07012345678"}
```

- Graf




- XML

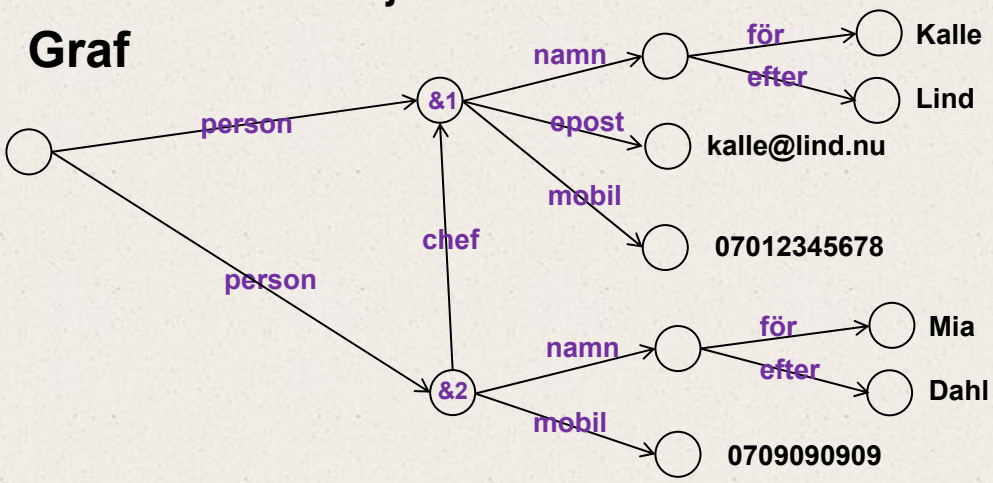
```
<Rot>
        <namn för="Kalle" efter="Lind" />
        <epost>kalle@lind.nu</epost>
        <mobil>07012345678</mobil>
</Rot>
```

# Träd vs. Nätverk

- **SSD**

```
{person: &1{namn:{för:"Kalle", efter:"Lind"},
            epost:"kalle@lind.nu",
            mobil:"07012345678"},
 person: &2{namn:{för:"Mia", efter:"Dahl"},
            mobil:"0709090909",
            chef: &1}
```

- **Graf**








# XML

- Står för Extensible Markup Language
- Ett språk för att definiera dokumentstrukturer
- XML är en textuell representation av data
- Används inom olika områden:
  - Datalagring
  - Webb (XHTML)
  - Konfigurationsfiler
  - Transportformat
- Regler kan specificeras via
  - DTD (Document Type Definition)
  - XML Schema
- Case sensitive

# XML-syntax

- **Element**
  ```
  <Person>Kalle</Person>
  ```
- **Attribut**
  ```
  <Person namn="Kalle"></Person>
  ```
- **Nästlade element**
  ```
  <Person id="59">
    <Fnamn>Kalle</Fnamn>
    <Enamn>Lind</Enamn>
    <Adress>
      <Gata>Kungsgatan 53</Gata>
      <Postnr>12332</Postnr>
      <Ort>Stockholm</Ort>
    </Adress>
  </Person>
  ```
- **Tomt element**
  ```
  <Person namn="Kalle"></Person>
  <Person namn="Kalle" />
  ```





# XML-dokument

- **XML-deklaration**
  ```
  <?xml version="1.1" encoding="UTF-8" ?>
  ```
- **DOCTYPE – referens till regler**
  ```
  <!DOCTYPE Person SYSTEM "Person.dtd">
  ```
- **Namespaces**
  - kvalificering av element- och attributnamn
    ```
    <iv1023:Person iv1023:namn="Kalle"></iv1023:Person>
    ```
  - default och andra namespaces
    ```
    <Root xmlns="default ns URI" xmlns:iv1023="iv1023 ns URI">
    …
    </Root>
    ```

# XML-referenser

- ID

```
<Person namn="Kalle" id="39"></Person>
```

- IDREF

```
<Organisation namn="KTH" chef="39"></Organisation>
```





# DTD (Document Type Definition)

- Definierar XML-strukturen (element och attribut)

```
<!ELEMENT db (Person*)>
<!ELEMENT Person (Adress)>
<!ELEMENT Adress EMPTY>
<!ATTLIST Person
        namn CDATA #REQUIRED
        id ID #REQUIRED
        fdatum CDATA #IMPLIED
        pappa IDREF #IMPLIED>
<!ATTLIST Adress
        gatuadress CDATA #REQUIRED
        postnr CDATA #REQUIRED
        postort CDATA #REQUIRED>
```

# XML Schema

- Starkare än DTD
  - flexiblare strukturer
  - datatyper
- XML-syntax

```
<element name="db" type="dbType"/>
<complexType name="dbType">
    <sequence>
        <element name="Person" type="PersonType" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="PersonType">
    <sequence>
        <element name="Adress" type="AdressType" />
    </sequence>
    <attribute name="namn" type="string" use="required"/>
    <attribute name="id" type="id" use="required"/>
    <attribute name="fdatum" type="date" use="optional"/>
    <attribute name="pappa" type="idref" use="optional"/>
</complexType>
<complexType name="AdressType">
…
```





# Well-formed & Valid

- Well-formed XML
  - Syntaktiskt korrekt
  - Börjar med XML-deklarationen
  - Innehåller endast ett rot-element
  - Matchade öppnings- och stängningstaggar
- Valid XML
  - Är well-formed
  - Följer reglerna i den kopplade DTD eller XML Schema

# XML-baserade språk

- Definition av struktur
- Definition av semantik

- XML-grundregler
  - Alfabet, vokabulär
- XML Schema (eller DTD)
  - Grammatik, syntax
- XML Schema förklaringen (för människor)
  - Semantik, betydelse





# XML-representation

- Textuell representation (serialiserat XML-dokument)
- Abstrakt nodstrukturrepresentation
  - XML Infoset
  - PSVI (Post-schema-validation Infoset)
  - XPath 1.0-modellen
  - XQuery 1.0-modellen
  - (XQuery 3.0.modellen)

# XML Infoset

- Representation av det väsentliga innehållet i ett XML-dokument
  - Vissa syntaktiska detaljer ignoreras
  - Bryr sig inte om XML Schema eller datatyper
- 11 information items, bl a
  - Document Information Item ("roten")
  - Element Information Item
  - Attribute Information Item
  - Comment Information Item
  - Processing Instruction Information Item
  - Document Type Declaration Information Item
  - Character Information Item
  - Namespace Information Item





# PSVI

- Post-Schema-Validation Infoset
- Utökar Infoset med stöd för information från XML Schema
  - datatyper
  - valideringstillstånd

# XPath 1.0-modellen

- Trädrepresentation av XML-dokument
- 7 nodtyper
  - root
  - element
  - attribute
  - text
  - namespace
  - comment
  - processing instruction
- Varje nod har ett värde
  - konkateneringen av alla underliggande textnoder
- Nodmängder (node sets)





# XQuery 1.0-modellen (XPath 2.0)

- Kan representera
  - XML-dokument (trädstruktur)
  - noder
  - värden
  - sekvenser av noder och/eller värden
- 7 typer av noder
  - document
  - element
  - attribute
  - text
  - comment
  - processing instruction
  - namespace

http://www.w3.org/TR/xpath-datamodel/

# XPath/XQuery 3.0-modellen

- Utökar föregående version med bl a
  - funktioner (även map och array)

https://www.w3.org/TR/xpath-datamodel-30/





# XQuery-modellen - nodegenskaper

- Elementnod
  - children (elementnoder, PI-noder, kommentarnoder, textnoder)
  - parent (elementnod eller dokumentnod)
  - attributes (attributnoder)
  - namespaces (namespacenoder)
  - string-value, typed-value
  - Obs! namespaces och attribut är inte children
- Attributnod
  - parent (elementnod) (heter owner i Infoset)
  - string-value, typed-value
- Dokumentnod
  - children
  - string-value, typed-value

# XQuery-modellen - nodegenskaper

- Textnod
  - string-value
  - typed-value
  - parent (elementnod)
- Kommentarnod
  - string-value
  - parent (elementnod eller dokumentnod)
- PI-nod
  - string-value
  - parent (elementnod eller dokumentnod)
- Namespace-nod
  - string-value
  - parent (elementnod)





# DTD (Document Type Definition)

- Definierar element och deras struktur
  - subelement
  - innehåll (content)
  - attribut
- Använder egen syntax

# DTD - element

- Tomt element
  <!ELEMENT Person EMPTY>
- Subelement
  <!ELEMENT Person (Namn, Adress)>
- Subelement som kan saknas
  <!ELEMENT Person (Namn, Adress?)>
- Subelement som förekommer noll eller flera gånger
  <!ELEMENT Person (Namn, Adress, Tel*)>
- Subelement som förekommer en eller flera gånger
  <!ELEMENT Person (Namn, Adress, Tel+)>
- Element med textinnehåll
  <!ELEMENT Person (#PCDATA)>
- Element med godtyckligt innehåll
  <!ELEMENT Person ANY>





# DTD - element

- Alternativa subelement
  <!ELEMENT Person (Namn, (Anställd|Student))>

Namn, Adress, Tel, Anställd, Student måste förstås också definieras.

# DTD - attribut

- Attribut definieras per elementtyp
- Ett eller flera attribut kan definieras i samma ATTLIST
- Obligatoriskt attribut
  <!ATTLIST Person pnr CDATA #REQUIRED>
- Frivilligt attribut
  <!ATTLIST Person längd CDATA #IMPLIED>
- Frivilligt attribut med default-värde
  <!ATTLIST Person födelseplats CDATA "Stockholm">
- Attribut med fast värde
  <!ATTLIST Person arbetsgivare CDATA #FIXED "KTH">
- Attribut med begränsade möjliga värden
  <!ATTLIST Person kön ("man", "kvinna") #REQUIRED>





# DTD - attribut

- Flera attribut i samma ATTLIST
  <!ATTLIST Person
  pnr CDATA #REQUIRED
  födelseplats CDATA "Stockholm"
  kön ("man", "kvinna") #REQUIRED
  längd CDATA #IMPLIED>
- Attribut med unika värden
  <!ATTLIST Person pnr ID #REQUIRED>
- Attribut som refererar ett unikt ID-attribut
  <!ATTLIST Person chef IDREF #IMPLIED>

# DTD – koppling till XML-dokument

- DOCTYPE
  - <!DOCTYPE Kurser SYSTEM "kurser.dtd">
- Inline
  - <!DOCTYPE Kurser [
    deklarationer ELEMENT, ATTLIST, etc.
    ]>





# XML Schema

- Definierar element och deras struktur
  - subelement
  - innehåll (content)
  - attribut
  - datatyper
  - antal förekomster
  - komplexa ordningsregler
- Använder XML-syntax
  - XML Schema är ett XML-baserat språk

# XML Schema-dokument

- Namespace
  - http://www.w3.org/2001/XMLSchema
  - rekommenderat alias: xs
- Rotelement
  - schema
  - attribut targetNamespace
- Definierar
  - element
  - attribut
  - typer

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        targetNamespace="…">
        definitioner
</schema>
```





# XML Schema-dokument

- Rotelementet schema
  - attribut elementFormDefault
    - » qualified
    - » unqualified (default)
  - attribut attributeFormDefault
    - » qualified
    - » unqualified (default)
  - Används för att bestämma om namespaces måste anges

```
<iv1023:Kurs xmlns:iv1023="http://ns.kth.se/IV1023">
        <Lärare namn="nikos"/>
</iv1023:Kurs>
<iv1023:Kurs xmlns:iv1023="http://ns.kth.se/IV1023">
        <iv1023:Lärare iv1023:namn="nikos"/>
</iv1023:Kurs>
```

Endast relevanta när man tänker blanda ihop flera namespaces och lokala element/attribut

# XML Schema - element

- Elementet element definierar element
  - attributet name definierar elementets namn
  - attributet type eller innehållet anger elementets typ
- Element kan ha en av följande typer
  - En grundtyp (string, integer, date, etc.)
    - » <element name="Namn" type="string" />
  - En typ definierad någon annanstans
    - » <element name="Namn" type="MinTyp" />
  - En typ definierad i innehållet
    - » <element name="Namn">
      typdefinitionen
      </element>





# XML Schema - attribute

- Elementet attribute definierar attribut
  - attributet name definierar attributets namn
  - attributet type eller innehållet anger attributets typ
  - attributet use anger om attributet är optional (default) eller required
- Attribute kan ha en av följande typer
  - En grundtyp (string, integer, date, etc.)
    - » <attribute name="Namn" type="string" />
  - En typ definierad någon annanstans
    - » <attribute name="Namn" type="MinTyp" />
  - En typ definierad i innehållet
    - » <attribute name="Namn">
      <simpleType … />
      </attribute>

# XML Schema - typer

- Grundtyper
  - string, integer, date, etc.
- Egendefinierade typer
  - complexType
    - » när man har subelement eller attribut
  - simpleType
    - » begränsning av en grunddatatyp





# XML Schema - simpleType

```
<xs:simpleType name="veckodag">
        <xs:restriction base="xs:string">
                <xs:enumeration value="Måndag"/>
                <xs:enumeration value="Tisdag"/>
                <xs:enumeration value="Onsdag"/>
                <xs:enumeration value="Torsdag"/>
                <xs:enumeration value="Fredag"/>
                <xs:enumeration value="Lördag"/>
                <xs:enumeration value="Söndag"/>
        </xs:restriction>
</xs:simpleType>

<xs:simpleType name="betyg">
        <xs:restriction base="xs:integer">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="10"/>
        </xs:restriction>
</xs:simpleType>
```

# XML Schema - complexType

- Kan innehålla ett av följande
  - simpleContent
    - » om elementets innehåll skall vara av en enkel datatyp (en textnod)
  - complexContent
    - » om elementets innehåll är baserat på en annan complexType
  - all
    - » definierar subelement i obestämd ordning
  - choice
    - » definierar alternativa subelement
  - sequence
    - » definierar sekvens av subelement
  - group
    - » använder en fördefinierad struktur av subelement
- Kan även innehålla (när inte Content)
  - noll eller flera element "attribute" eller "attributeGroup"



IV1023 vt2022
nikos dimitrakas
KTH/EECS/SCS

# XML Schema - complexType

```
<xs:complexType name="PersonType">
    </xs:sequence>
        <xs:element name="Förnamn" type="xs:string"/>
        <xs:element name="Efternamn" type="xs:string"/>
        <xs:element name="Födelsedatum" type="xs:date"/>
    </xs:sequence>
</xs:complexType>


<xs:complexType name="PersonType">
    <xs:attribute name="Förnamn" type="xs:string"/>
    <xs:attribute name="Efternamn" type="xs:string"/>
    <xs:attribute name="Födelsedatum" type="xs:date"/>
</xs:complexType>
```

# XML Schema - förekomster

- Attributet minOccurs anger minsta antal förekomster
  - default 1
- Attributet maxOccurs anger högsta antal förekomster
  - default 1
  - för obegränsat sätt "unbounded"

- Används för
  - choice
  - all (0..1 till 1..1)
  - sequence
  - group
  - element
  - any (godtyckligt element)





# XML Schema - exempel

```
<xs:group name="InfoGrupp">
    </xs:choice>
        <xs:element name="Epost" type="xs:string"/>
        <xs:element name="Telefon" type="xs:string"/>
        <xs:element name="Webbplats" type="xs:string"/>
    </xs:choice>
</xs:group>

<xs:complexType name="PersonTyp">
    <xs:sequence>
        <xs:group ref="InfoGrupp" minOccurs="0" maxOccurs="unbounded" />
        <xs:element name="Förnamn" type="xs:string"/>
        <xs:element name="Efternamn" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="Födelsedatum" type="xs:date" use="optional"/>
</xs:complexType>

<xs:complexType name="PersonalTyp">
    <xs:complexContent>
        <xs:extension base="PersonTyp">
            <xs:attribute name="Anställningsnummer" type="xs:string"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

# XML Schema - Övrigt

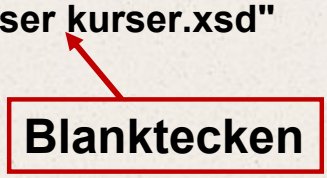
- list
  - definierar att en simpleType är en lista
- union
  - definierar en ny typ som är unionen av flera andra typer (simpleTypes)
- annotation
  - för att ange kommentarer (med documentation eller appInfo)
- key, unique, field, keyref, selector
  - definierar "identifierare"/"kandidatnycklar" och referenser
- datatypbegränsningar med
  - length, minLength, maxLength
  - minExclusive, minInclusive, maxExclusive, maxInclusive
  - totalDigits, fractionDigits
  - pattern, whiteSpace





# XSD - Koppling till XML-dokument

- Med Namespace
  - Ett attribut (från XSI-namespace) i rotelementet och namespace-definition (default eller med prefix)
  - xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  - xsi:schemaLocation="http://ns.kth.se/IV1023/kurser kurser.xsd"
  - xmlns="http://ns.kth.se/IV1023/kurser"
    eller
    xmlns:k="http://ns.kth.se/IV1023/kurser"




- Utan Namespace
  - Ett attribut (från XSI-namespace) i rotelementet:
  - xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  - xsi:noNamespaceSchemaLocation="kurser.xsd"

# Fortsättning

- Quiz om semistrukturerade data, XML, DTD, XML Schema