SDXML VT2024
Models and languages for semi-structured data and XML

Introduction to the course Semi-structured data and XML

nikos dimitrakas
nikos@dsv.su.se
08-161295

Corresponding reading
Excerpt from Data on the Web
Chapter 1, 4, 5, 6, 10 (especially 10.6) of the course book
Parts of chapter 30 of Database Systems (Connolly, Begg) 6th edition (chapter 31 in 5th edition)

Course content

- **Semi-structured data, XML and JSON**
  - **Data**
  - **Model (DTD, XML Schema, JSON Schema)**
  - **Representation**
  - **Usages (Open data, XML-based languages)**

Semi structured data, XML and JSON
Data
Model(DTD,XML schema ,JSON schema)
Representation
Usages(open data, XML based languages)

- **Query languages**
  - **Lorel**
  - **XPath**
  - **XQuery**
  - **XSL/XSLT**
  - **SQL/XML (part of SQL 2003)**
  - **Product-specific techniques (IBM, Oracle, Microsoft)**

Query languages
Lorel
Xpath
Xquery
XSL/XSLT
SQL/XML
IBM,Oracle,Microsoft

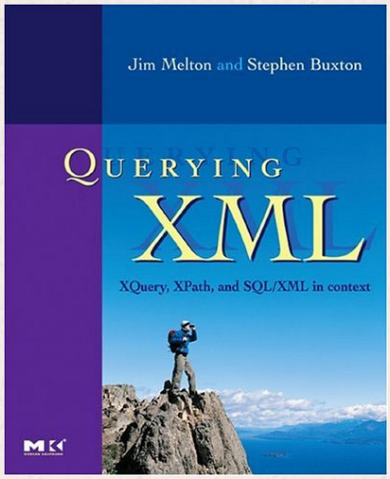
Course setup

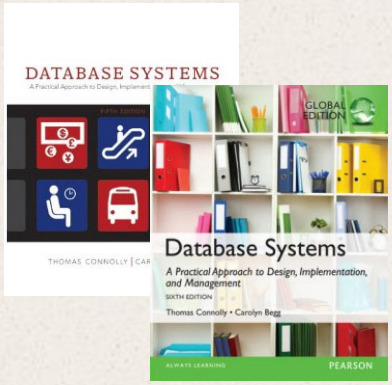
- Lectures
- Lessons
- Quizzes
- Seminars with submission
- Tutorials (Introduction to technologies and products)
- Assignments
- Tutoring
- Exam
- Feedback

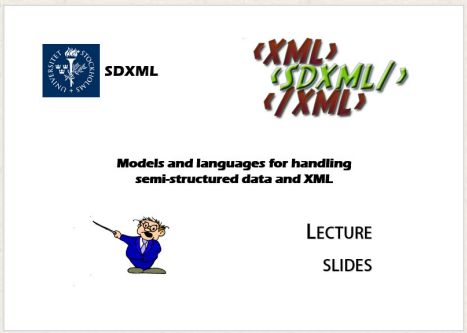
Material

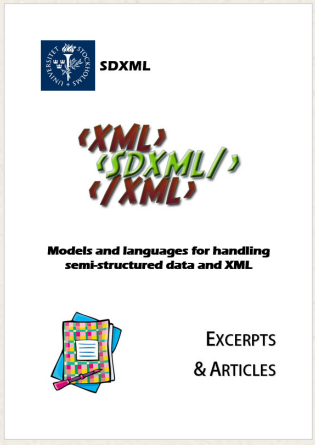
- Course information compendium
- Lecture slides (only electronically)
- Compendiums about the technologies and the products (only electronically)
  - Tutorials
- Books
  - Basic database book (Database Systems, Connolly/Begg, edition 6)
  - Course book (Querying XML, Melton/Buxton)
  - Other XML books
- Excerpts and articles (only electronically)
- Other material
  - Relevant web pages
  - Suggested solutions to the lesson exercises
  - Sample databases
  - Sample/Old exams

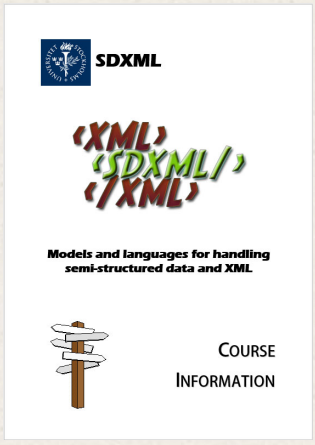
Material











Examination

- **Examination 1 (3,5 hec) F-A (XMLT in Ladok)**
  - **Written exam**
- **Examination 2 (2,5 hec) U-G (XML1 in Ladok)**
  - **Quizzes**
  - **Assignments 1-5**
  - **Seminars (assignments 1-3)**
- **Examination 3 (1,5 hec) F-A (XML2 in Ladok)**
  - **Assignments 6-12**
    - » **9-12 optional for D, C, B, A**
- **”Examination” 4 (0 hec)**
  - **Course evaluation**

Course information compendium

- General information
  - Literature, teachers, activities, software, tutoring, examination, evaluation…
- Suggested workflow
- Lessons exercises
- Assignments
- Quizzes
- Sample databases

Read through it!

Software

- Administration and communication
  - Daisy
  - iLearn
  - The tutoring system
- Exercises, Tutorials, Assignments
  - Database Management Systems
    - » Oracle 19c
    - » DB2 11.5
    - » SQL Server 2019
  - XQuery
    - » XQuisitor
    - » BaseX
  - XSLT
    - » Web browsers
    - » Web sites xsltransform.net, xslttest.appspot.com
  - XML, JSON
    - » Validation web sites
    - » Notepad++

Xquery run on Xquisitor and BaseX

XSLT - web browsers

Groups

- **Assignments**
  - 1-8 in groups of 3 (2 if necessary)
  - 9-11 in groups of 1-3
  - 12 individually
- **Quizzes**
  - individually
- **Form groups in iLearn**
  - Use forum in iLearn if necessary

End of introduction to the course

Data - Metadata

- **Data**
  - **Johnny, Pasta, Lund, 2001-02-12, true, 677**
- **Metadata**
  - **name, name, city, start date, sent, weight**
- **Types of metadata**
  - **Structural**
  - **Semantic**
  - **Catalog (classification)**
  - **Integration (mapping)**

Data - metadata
Johnny, Pasta, Lund
Metadata - name,city,start date,sent,weight

Types of metadata
Structural metadata
Semantic metadata
Catalog metadata
Integration metadata

Structure

- **Modeling**
  - 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.
- **Database solutions**
  - **Relational model**
    - » **Tables, columns, domains, keys, integrity constraints**
  - **Object-oriented, Object databases**
    - » **Classes, attributes, references, rules**
  - **XML**
    - » **Elements, attributes, rules**
  - **Other**
    - » **?**

data modeling - analysis of data objects used in business or other contexts and identification of relationships among these objects.

Relational model - Tables ,columns, domains, keys, integrity constraints

XML - elements,attributes,rules

Semantics

- **The meaning of the data and metadata**
  - **Metadata**
    - » **name**
    - » **price**
    - » **weight**
    - » **sent**
  - **Semantics**
    - » **The thing that identifies each product type uniquely**
    - » **The number of SEK the customer pays including VAT for one piece**
    - » **Specifies the weight of the product including packaging in grams**
    - » **True if the order has been sent from our storage, otherwise false**

semantics - meaning of data and metadata

metadata - name, price, weight , sent

Semantics
What identifies each product type uniquely
number of SEK customer pays including VAT for one piece

Semi-structured data

- **No structure (schemaless)**
- **Implicit structure (self-describing)**
  - **metadata built-in to the data**
    - » **no data → no metadata**
- **SSD**

```
{name:{first:"Kalle", last:"Lind"},
 email:"kalle@lind.nu",
 mobile:"07012345678"}

{name:"Lisa",
 phone:"0709999999"}
```

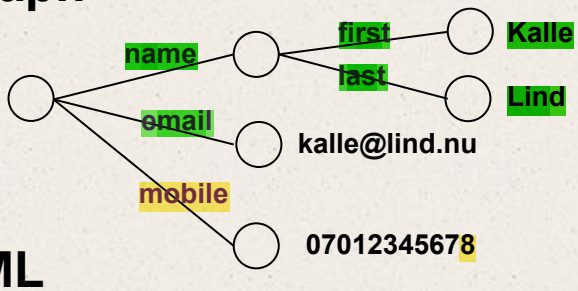
semi structured data
no structure
implicit structure

Representations

- SSD

```
{name:{first:"Kalle", last:"Lind"},
 email:"kalle@lind.nu",
 mobile:"07012345678"}
```

- Graph



- XML

```
<Root>
        <name first="Kalle" last="Lind" />
        <email>kalle@lind.nu</email>
        <mobile>07012345678</mobile>
</Root>
```

- JSON

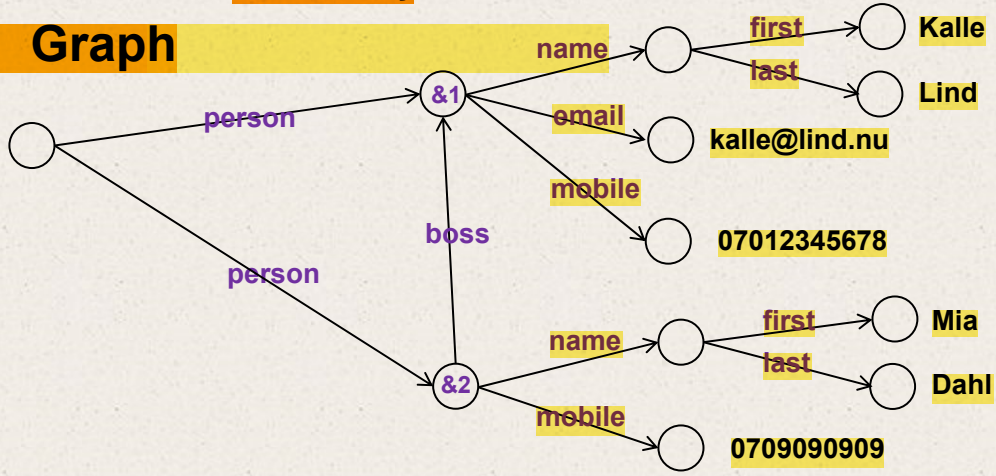
```
{"name" : {"first" : "Kalle",
           "last" : "Lind"},
 "email" : "kalle@lind.nu",
 "mobile" : "07012345678"}
```


Tree vs. Network

- SSD

```
{person: &1{name:{first:"Kalle", last:"Lind"},
            email:"kalle@lind.nu",
            mobile:"07012345678"},
 person: &2{name:{first:"Mia", last:"Dahl"},
            mobile:"0709090909",
            boss: &1}
```

- Graph



XML

- Stands for Extensible Markup Language
- A language for defining document structures
- XML provides a textual representation of data
- Is used within different areas:
  - Data storage
  - Web pages (XHTML)
  - Configuration files
  - Transport format (integrations, conversions)
- Rules can be specified through
  - DTD (Document Type Definition)
  - XML Schema
- Case sensitive

XML - extensible Markup Language
Language for defining document structures
Provides textual representation of data

Used within different areas
Data storage
Web pages
Configuration files
Transport format

Rules specified through
-DTD or XML schema
Case sensitive.

XML - Syntax

- Element

```
<Person>Kalle</Person>
```

- Attribute

```
<Person name="Kalle"></Person>
```

- Nested elements

```
<Person id="59">
    <Fname>Kalle</Fname>
    <Lname>Lind</Lname>
    <Address>
        <Street>Kungsgatan 53</Street>
        <PostalCode>12332</PostalCode>
        <City>Stockholm</City>
    </Address>
</Person>
```

- Empty element

```
<Person name="Kalle"></Person>
<Person name="Kalle" />
```

Empty element - nothing inside the element

XML syntax
Element
<Person>Kalle</Person>

Attribute
<Person name="Kalle"></Person>

Nested elements
<Person id="59">
<Fname>Kalle </Fname>
<Lname>Lind</Lname>

<Address>
<Street>
<Postal Code>
<City>

</Address>
</Person>

XML Document

- **XML declaration**

  <?xml version="1.1" encoding="UTF-8" ?>

- **DOCTYPE – reference to rules**

  <!DOCTYPE Person SYSTEM "Person.dtd">

- **Namespaces**

  – **qualification of element and attribute names**

  <sdxml:Person sdxml:name="Kalle"></sdxml:Person>

  – **default and other namespaces**

  <Root xmlns="default ns URI" xmlns:sdxml="sdxml ns URI">

  …

  </Root>

declaration has version and encoding.

<?xml version="1.1" encoding="UTF-8"?>

<!DOCTYPE Person SYSTEM "
Person.dtd">

Namespaces
qualification of element and attribute names
<sdxml:Person sdxml:name="Kalle">
</sdxml:Person>

XML - References

- **ID**

  <Person name="Kalle" id="39"></Person>

- **IDREF**

  <Organization name="IBM" boss="39"></Organization>

DTD (Document Type Definition)

- **Defines the XML structure (elements and attributes)**

```
<!ELEMENT db (Person*)>
<!ELEMENT Person (Address)>
<!ELEMENT Address EMPTY>
<!ATTLIST Person
        name CDATA #REQUIRED
        id ID #REQUIRED
        birthdate CDATA #IMPLIED
        father IDREF #IMPLIED>
<!ATTLIST Address
        street CDATA #REQUIRED
        code CDATA #REQUIRED
        city CDATA #REQUIRED>
```

DTD
Defines the XML structure(elements and attributes)

<!ELEMENT db(Person*)>

XML Schema

- **Stronger than DTD**
  - **More flexible structures**
  - **data types**
- **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="Address" type="AddressType" />
    </sequence>
    <attribute name="name" type="string" use="required"/>
    <attribute name="id" type="id" use="required"/>
    <attribute name="birthdate" type="date" use="optional"/>
    <attribute name="father" type="idref" use="optional"/>
</complexType>
<complexType name="AddressType">
…
```

Well-formed & Valid

- **Well-formed XML**
  - **Syntactically correct**
  - **Starts with an XML declaration**
  - **Contains only one root element**
  - **Matching opening and closing tags**
- **Valid XML**
  - **Is well-formed**
  - **Follows the rules of the associated DTD or XML Schema**

XML-based languages

- **Definition of structure**
- **Definition of semantics**

- **XML basic rules**
  - **Alphabet, vocabulary**
- **XML Schema (or DTD)**
  - **Grammar, syntax**
- **XML Schema explanation (for humans)**
  - **Semantics, meaning**

XML - Representations

- Textual representation (serialized XML document)
- Abstract node structure representation
  - XML Infoset
  - PSVI (Post-schema-validation Infoset)
  - XPath 1.0 model
  - XQuery 1.0 model
    - » XQuery 3.0 model
    - » XQuery 3.1 model

XML Infoset
PSVI - Post schema validation Infoset
Xpath 1.0
Xquery 1.0
Xquery 3.0 model
Xquery 3.1 model

XML Infoset

- Representation of the significant parts of the content of an XML document
  - Some syntactical details are ignored
  - Does not care about XML Schema or data types
- 11 information items, among them
  - Document Information Item ("the root")
  - Element Information Item
  - Attribute Information Item
  - Comment Information Item
  - Processing Instruction Information Item
  - Document Type Declaration Information Item
  - Character Information Item
  - Namespace Information Item

XML infoset
Representation of significant parts of XML document.

PSVI

- Post-Schema-Validation Infoset
- Extends Infoset with support for XML Schema information
  - data types
  - validation status

PSVI
Post schema validation infoset

XPath 1.0 model

- Tree representation of XML documents
- 7 node types
  - root
  - element
  - attribute
  - text
  - namespace
  - comment
  - processing instruction
- Every node has a value
  - The concatenation of all contained text nodes
- Node sets

Xpath - tree representations of XML documents
root
element
attribute
text
namespace
comment
processing instruction

1999

XQuery 1.0 model (XPath 2.0)

- **Can represent**
  - **XML documents (tree structure)**
  - **nodes**
  - **values**
  - **sequences of nodes and/or values**

Can represent
XML documents
nodes
values
sequences of nodes and/or values

- **7 types of nodes**
  - **document**
  - **element**
  - **attribute**
  - **text**
  - **comment**
  - **processing instruction**
  - **namespace**

7 types of nodes
document
element
attribute
text
comment

2007

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

XPath/XQuery 3.0 model

- **Extends the previous version with**
  - **functions**

2014

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

XPath/XQuery 3.1 model

- Extends the previous version with
  - maps
  - arrays

2017

https://www.w3.org/TR/xpath-datamodel-31/

XQuery model - node properties

- Element node
  - children (element nodes, PI nodes, comment nodes, text nodes)
  - parent (element node or document node)
  - attributes (attribute nodes)
  - namespaces (namespace nodes)
  - string-value, typed-value
  - Namespaces and attributes are not children
- Attribute node
  - parent (element node) (called owner in Infoset)
  - string-value, typed-value
- Document node
  - children
  - string-value, typed-value

XQuery model - node properties

- Text node
  - string-value
  - typed-value
  - parent (element node)
- Comment node
  - string-value
  - parent (element node or document node)
- PI node
  - string-value
  - parent (element node or document node)
- Namespace node
  - string-value
  - parent (element node)

What to do next

- Quiz about XML (Quiz 1)