Høgskulen
på Vestlandet

Databaser

DOM, XLST, Xpath, Query,
and XML in/ from databases

Dag Larsen
Oppdatert av Lasse Jenssen

Plan for NoSQL delen

- Video 1: Introduksjon til NoSQL
- XML og JSON (Kap. 14)
  - Video 2: XML, DTD og XML Schema
  - **Video 3: DOM, XSLT, XPath, XQuery, XML i PostgreSQL**
  - Video 4: JSON, JSON i PostgreSQL
- Via objekter (objekt relasjonelle databasesystemer) til NoSQL databaser (Kap. 15)
  - Video 5: Via objekter til NoSQL databaser
  - Video 6: NoSQL databaser
- NoSQL og MongoDB (Kap. 15.4)
  - Diverse videoer (kommer tilbake til disse etter hvert)
  - Vi bruker Docker eller MongoDB Atlas (eventuelt lokal installasjon på PC)
- Obligatorisk Innlevering 4 (legges ut i starten av uke 15)

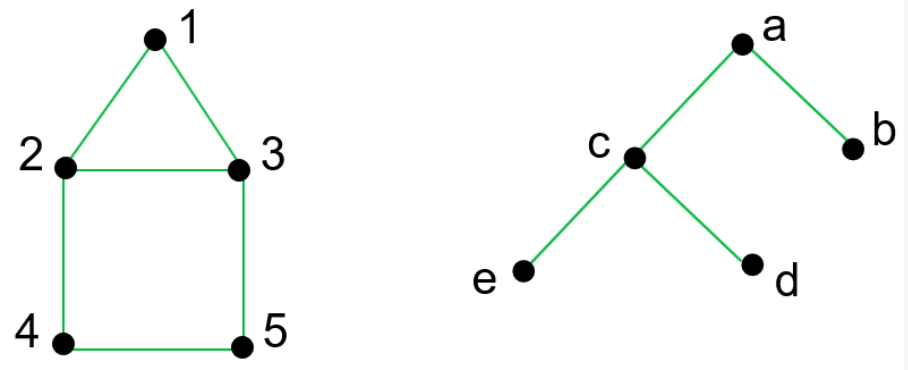
Læringsmål

- Forstå hva menes med **DOM (Document Object Model)**
- Forstå enkel bruk av **stilark (XSLT)**
- Forstå enkel bruk av **spørrespråk for XML (XPath og XQuery)**.

Graf (til venstre) og tre (til høyre)

- «graf» (figur 1): Finner vi i nettverksdatabaser
- «Trestrukturer» (figur 2): kan vi for eksempel finne i ...
  - Relasjonsdatabaser (En-til-mange egenforhold)
  - Dokument databaser (XML eller JSON)
- XML-dokumenter er trestrukturer
  - Forståelsen av XML som trestruktur er nyttig for å beskrive lovlige dokumenter, for å referere til deler av et XML-dokument og for å lage stilark for XML

Figur 1

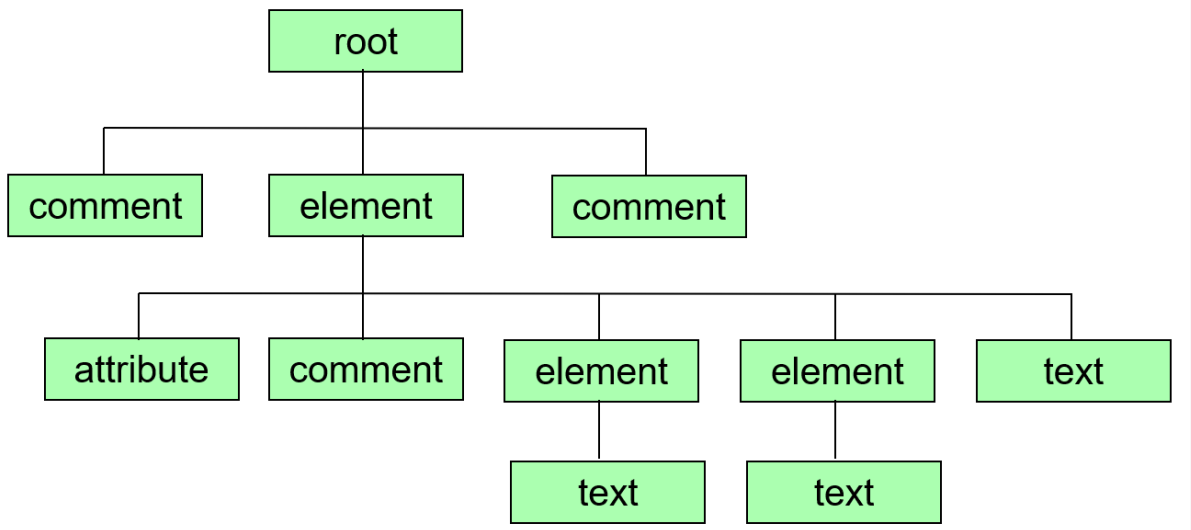


Figur 2

DOM (Document Object Model)

DOM (Document Object Model) er en trestruktur av objekter

- Kan behandles med forskjellige programmeringsspråk
- Metoder: addChild, getChild, getSibling, …
- Definert både for HTML og XML



Nodetyper i DOM

- Det er 7 nodetyper i et XML-tre:
  - element
  - attribute
  - text
  - namespace
  - processing-instruction
  - comment
  - document (root) nodes

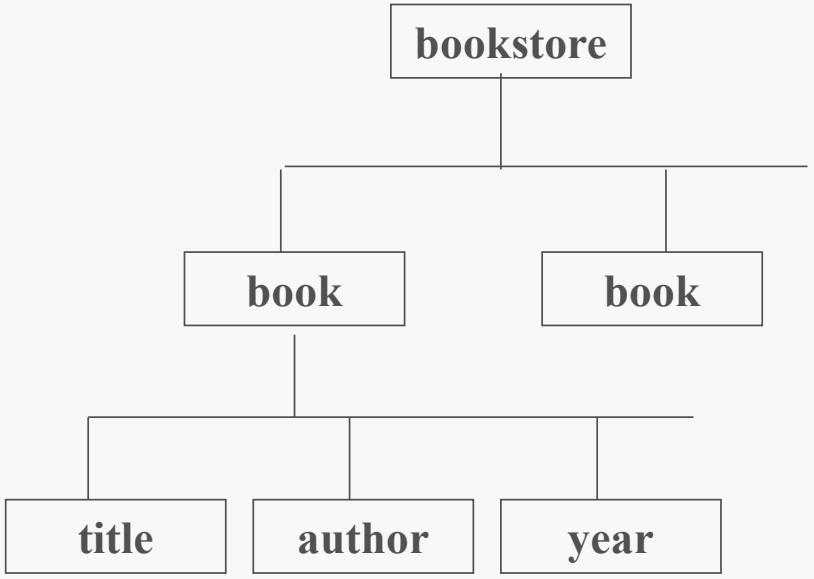
XPath

- For å lage regler som «skriver om» eller å «håndtere» XML-dokumenter er det nyttig å kunne snakke om deler av et XML-dokument
  - Eksempel: Innholdet av elementet <title> som befinner seg under første forekomst av elementet <book>

- **XPath** er en notasjon for å referere til deler av et XML-dokument
  - XPath er basert på **stiuttrykk**
  - XPath inneholder et bibliotek av standardfunksjoner
  - XPath er en W3C Standard

Trestrukturen

- Foreldre, barn, søsken
- Forfedre, avkom



XPath stiuttrykk

- Eksempel:

```
<tavle>
  <melding mld="SQL ruler. Lenge leve NoSQL.">
     <avsender aId="1001">
        <fornavn>Kari</fornavn><etternavn>Lie</etternavn>
     </avsender>
     <mottaker mId="1002">
        <fornavn>Ola</fornavn><etternavn>Hansen</etternavn>
     </mottaker>
  </melding>
</tavle>
```

Stiuttrykk: **/tavle/melding/avsender**

XPath stiuttrykk

/tavle/melding	Alle meldinger direkte under tavle.
/tavle/melding[3]	Den tredje meldingen direkte under tavle.
/tavle//etternavn	Alle etternavn under tavle (direkte eller indirekte).
/tavle//*	Alle elementer under tavle.
/tavle/melding@mId	Alle mId-attributter til meldinger direkte under tavle.
/tavle//etternavn..	Foreldrenoder til etternavn under tavle.
avsender[@aId=1002]	Avsendere med attributt aId lik 1002 direkte under aktiv node.
text()	Alle text-noder direkte under aktiv node.

XSLT

- XSL = eXstensible Stylesheet Language

- XSLT = **stilark** for XML
  - XSLT = XSL Transformations
  - skriver om (transformerer) et XML-dokument til et annet XML-dokument – eller HTML (eventuelt annet)
  - bruker XPath for å navigere i XML-dokumenter

- Nettsider
  - Lær-selv-sider: www.w3schools.com/xsl
  - Standard/spesifikasjon: www.w3.org/Style/XSL/

Presentere XML med XSLT

- Med XSLT kan vi lage regler for å gjøre om XML til HTML
- Generelt format
  - XML-mønster => HTML + «XML-utplukk»
  - Høyresiden av regelen består av HTML ispedd kode for å plukke ut elementer fra XML-dokumentet
- Kan gjøre strukturelle endringer
  - Kan velge ut noen elementer
  - Kan sortere elementer
- Hvordan brukes XPath i XSL?
  - Brukes som mønster i venstresiden av regler
  - Brukes også for å plukke ut elementer i høyresiden

Eksempel på XSLT-regel

- Plukk ut <pnr>, <fornavn>, <etternavn> og <fodselsdato> fra <person>.
- Sett “Person - <pnr>” som overskrift (h2).
- Sett følgende i avsnitt (p):
  - Navn: fornavn og etternavn med et blankt tegn mellom
  - Fødseldato (med komma foran)

Person - 1

Navn: Ola Nordmann, Fødselsdato: 1980-01-01

Person - 2

Navn: Kari Svenske, Fødselsdato: 1993-08-23

Eksempel på XSLT-regel

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <xsl:output method="html"/>

    <xsl:template match="//person">
        <h3>Person - <xsl:value-of select="pnr"/></h3>
        <p>Navn: <xsl:value-of select="fornavn"/>
                 <xsl:text> </xsl:text>
                 <xsl:value-of select="etternavn"/> </p>
        Fødselsdato: <xsl:value-of select="fodselsdato"/><br />
    </xsl:template>

</xsl:stylesheet>
```

Et annet eksempel på XSLT-regel

- Legg navn (fornavn og etternavn), fødselsdato, og antall barn inn i en HTML tabell

Personer

Pnr	Navn	Fødseldato	Antall barn
1	Ola Nordmann	1980-01-01	1
2	Kari Svenske	1993-08-23	

Eksempel på XSLT-regel

```
<xsl:template match="personer">
    <HTML>
        <BODY>
            <h1>Personer</h1>
            <TABLE border='1' style='table-layout:fixed' width='600'>
                <TR bgcolor='#FFFF00'>
                    <TD><b>Pnr</b></TD><TD><b>Navn</b></TD>
                    <TD><b>Fødseldato</b></TD><TD><b>Antall barn</b></TD>
                </TR>
                <xsl:for-each select="person">
                    <TR>
                        <TD><xsl:value-of select='pnr' /></TD>
                        <TD><xsl:value-of select="fornavn"/>
                            <xsl:text> </xsl:text>
                            <xsl:value-of select="etternavn"/></TD>
                        <TD><xsl:value-of select='fodselsdato' disable-output-escaping="yes"/></TD>
                        <TD><xsl:value-of select='barn' disable-output-escaping="yes"/></TD>
                    </TR>
                </xsl:for-each>
            </TABLE>
        </BODY>
    </HTML>
</xsl:template>
```

XML og databaser

- Anvendelser
  - XML for dynamiske data til web
  - XML som overføringsformat
- Lagre XML i databaser
  - CLOB eller XML Type (Oracle: XMLType, PostgreSQL: XML)
- Spørringer mot XML-data
  - XPath
  - XQuery
- Produsere XML fra tabelldata

XQuery

- **“XQuery is to XML what SQL is to databases”**
- **Et funksjonelt programmering -og spørre språk** (bruker Xpath)
- Brukes til å trekke ut og manipulere/håndtere data fra enten XML documenter eller relasjonsdatabaser

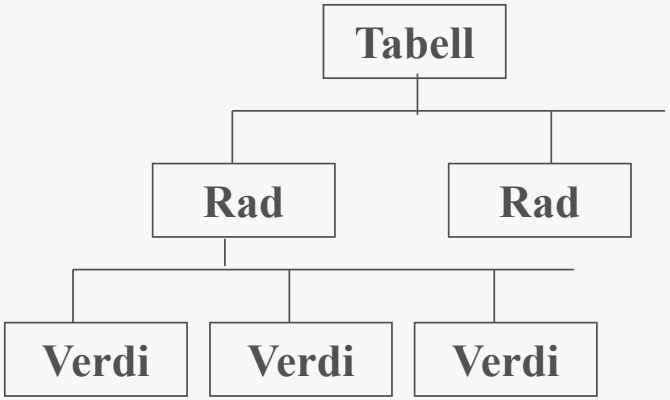
```
let $personer := (doc("personer.xml")/personer/person)

return <results>
{
   for $x in $personer
   where $x/barn>0
   order by $x/barn
   return $x/fornavn
}
</results>
```

https://www.w3.org/TR/xquery-31/

Strukturerte data i XML variant 1

- XML-representasjon av radsamlinger
  - Rot-element = tabellnavn
  - Elementnavn = navn på databasekolonne
  - Tekst i elementet (mellom taggene) = verdi

```
<?xml version="1.0" encoding="UTF-8"?>
<Studenter>
  <Student>
    <Studentnr>012345</Studentnr>
    <Fornavn>Kari</Fornavn>
    …
  </Student>
  <Student>
    <Studentnr>012346</Studentnr>
    …
  </Student>
</Studenter>
```



Strukturerte data i XML variant 2

- Alternativ: Data i attributter

```
<Student Studnr="012345", Fornavn=«Kari",
         Etternavn="Nordmann", Gate="Inndalsveien",
         Husnr="28", Postnr="5063",
         Poststed="Bergen">
</Student>
```

  - Elementnavn = tabellnavn
  - Attributtnavn = navn på databasekolonne
  - Attributtverdi = verdi

Semistrukturerte data

- Karakteristikk
  - Ikke tabellstruktur, mer en **dokumentstruktur**
  - Skjemafrie (følger ikke gitt typestruktur)
- Behov
  - Ønsker å lagre slike data i «databaser»
  - Ønsker å kunne søke (ad hoc) med et spørrespråk
- Hvordan lagre semistrukturerte data?
  - Som XML-dokumenter i filer
  - Normalisert tabellstruktur i relasjonsdatabase
  - Hybridløsning: XML-dokumenter som verdier i relasjonsbaser
  - Objektrelasjonelle databaser med XML-støtte
  - XML-databaser
  - NoSQL-databaser

XML-databaser

- Problemstillinger:
  - Er XML egnet for permanent lagring av data?
  - Kan XML-dokumenter erstatte databasesystemer?

- Hva kreves av en XML-database?
  - Kunne beskrive logisk datastruktur i XML med bl.a.:
    - Lovlige elementer og attributter
    - Datatyper og elementstruktur
    - Primærnøkler og fremmednøkler, forretningsregler
  - Et XML-basert spørrespråk for søking i XML-data
  - Mulighet for å oppdatere XML data

XML Type (PostgresDB)

- En egen datatype XML for å lagre XML-dokumenter:

```
CREATE TABLE personer (
  person_no     INTEGER,
  xml_data      XML
);
```

- Sette inn XML-data:

```
INSERT INTO personer (person_no, xml_data)
  VALUES ( 1,XMLPARSE( DOCUMENT '<person>
                  <fornavn>Ola</fornavn><etternavn>Nordmann</etternavn>
                  <barn>1</barn><fodselsdato>1980-01-01</fodselsdato>
                </person>'));
```

- For PostgreSQL-spesifikk info, se f.eks.:
  PostgreSQL: Documentation: 14: 8.13. XML Type
  PostgreSQL: Documentation: 14: 9.15. XML Functions

Datatypen XML (PostgreSQL)

- Nyttige operasjoner på XML type:
  - XPATH
  - XPATH_EXISTS
  - Ingen tilsvarende metode som UPDATEXML, DELETEXML (som finnes i Oracle)

- Hente ut et subtre:

```
SELECT person_id,
       XPath('/person/fornavn/text()', person_data) fornavn,
       XPath('/person/etternavn', person_data) etternavn,
       XPath('/person/barn/text()', person_data) barn
FROM personer
WHERE xpath_exists('/person[fornavn="Nils"]', person_data);
```

- Oppdatere deler av et XML-dokument (må hente ut xml, og manipulere xml med programkode)

```
UPDATE personer
   SET person_data = '<person><pnr>003</pnr><fornavn>Nilsi</fornavn>…</person>'
   WHERE person_id = 3;
```

XML ut fra databaser («vanlige tabeller») (PostgreSQL)

- Funksjoner for å «merke opp» et spørreresultat

```
SELECT XMLElement(name "fornavn", p.fornavn) navn
FROM personer p;

navn
-----------------------
<fornavn>Ola</fornavn>
<fornavn>Kari</fornavn>
   ....
SELECT XMLForest(p.person_no, p.fornavn, p.etternavn)  person
FROM personer p
WHERE p.id = 1;

person
-------------------------------------
<person_no>1</person_no><fornavn>Ola</fornavn><etternavn>Nordmann</etternavn>
```

XML for dataoverføring

- Hvorfor bruke XML som dataoverføringsformat?
  - Fleksibelt: Utvidbar syntaks
  - Selvbeskrivende (elementnavn)
  - «Åpent»: Lesbart for maskiner og mennesker (plattform uavhengig)
  - Gode verktøy for å automatisere arbeidet

- Men ...
  - Det finnes mange andre gode formater
  - Inneholder mye «unødige» data som tar plass
  - «Selvbeskrivende» er en sannhet med modifikasjoner
  - Mennesker kan gjette betydningen av data basert på elementnavn - det kan ikke maskiner

Oppsummering

- Transformasjon og presentasjon: XSLT
- XML og databaser
  - Strukturerte og semistrukturerte data
  - Lagring av XML-data i databaser
  - XML som overføringsformat
  - Spørrespråk mot XML (XPath og XQuery)
- XML i PostgreSQL
  - XPATH
  - XPATH_EXIST
- XML ut fra PostgreSQL
  - XMLELEMENT
  - XMLFORREST