6.3 Resource Description Framework (RDF)

6 Interoperabilität im WWW mit XML

6.1 XML – Definition und Nutzung

6.2 Anfragen auf XML-Datenbeständen

Einführung

- Beispiel "Elektronischer Marktplatz"
  - Anwendung: PC-Fertigung / Einkauf
  - Ziel: Produktionsplanung,
    Bauteile für Planungszeitraum unter Kosten- u. Qualitätsrestriktionen bereitstellen
  - Vorgehen für Bauteile ohne vorab festgelegte Lieferanten (z.B. RAM)
  - Beispiel: Suche 5000 Speicherbausteine à 8x32 Mbit,
    Zugriffszeit höchstens 8 nsec,,.....
  - Alternativen:
    - manuelle Suche mit Webbrowser ???
    - Suchmaschine auf textlicher Basis ??
    - Suchmaschine, die verteilte XML-Dokumente abfragt?

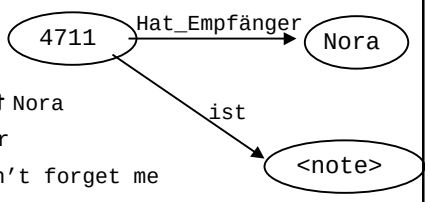
Ordnung im Chaos?

- Wie macht man beliebige Internet-Ressourcen (Programmen) zugänglich?
  - Standards
    - Conditio sine qua non (not just nice to have!)
    - Klar, aber man kann nicht alles standardisieren (Produktbeschreibungen, Infos, ...)
  - Gemeinsame Syntax und Semantik
  - Elementare Approximation einer semantischen Beschreibung: Wörter → Suchmaschine
  - XML ?
    - Nur Syntax – Kommunikationspartner müssen sich über gemeinsame Sprache verständigt haben
    - Syntaktische, keine semantische Konformität

Beispiel

```
<?xml version="1.0"?>
<note ID="4711">
   <to>Nora</to>
   <from>Chris</from>
   <heading>Reminder</heading>
   <body>Don't forget me this weekend!</body>
</note>
```

- "Bedeutung"
  - eine Nachricht `<note>`
  - Absender ist `Chris`
  - EmpfängerIn der Nachricht ist `Nora`
  - Nachricht hat Titel `Reminder`
  - Inhalt der Nachricht ist `Don't forget me this weekend!`



und XML?

- Syntax von RDF ist XML! (Serialisierung des Graphen)
- RDF: Darstellung von Fakten (Aussagen)

  `hat _Empfänger (4711, Nora)`

  `Prädikat (Subjekt, Objekt)`

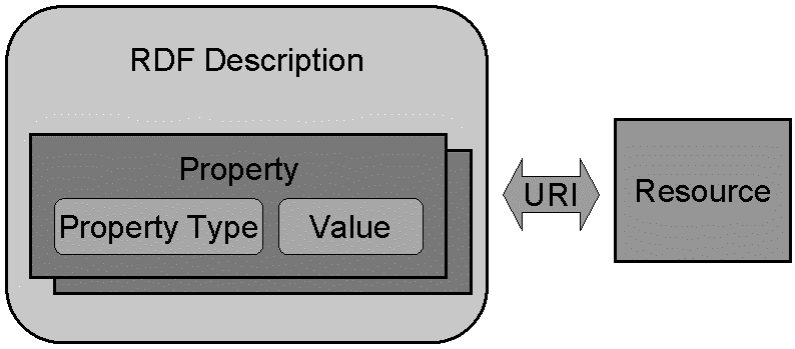
  in normierter Sprache
- Warum nicht gleich XML?
- Größere Beliebigkeit der XML-Syntax

```
<note to="Nora" from="Chris" heading="Reminder"
  body="Don't forget me this weekend!">
 </note>
```

Extremfall, aber korrekte XML-Syntax!
Allerdings: wer so etwas schreibt, hat's nicht verstanden!

6.5

Das Modell von RDF



6.6

Resource Description Framework

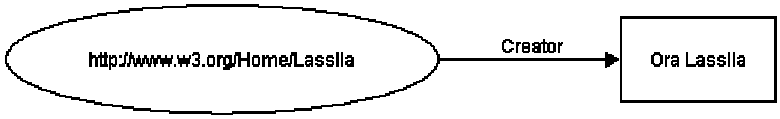
- RDF description
  - Beschreibung einer beliebigen resource die durch URI (uniform resource identifier) identifiziert werden kann
    - Bsp.: Web Server, Dokument, Bild, ...
  - ... durch Eigenschaften (properties)
  - Eigenschaften haben als Wert (value) Literale oder Resourcen
    Literale eigentlich Spezialfall von Resourcen

- RDF description = {(Resource , property, Value)}

  oder: (Subject, predicate, Object)

6.7

Beispiel



Subjekt (Resource) `http://www.w3.org/Home/Lassila`
Prädikat: `Creator`
Objekt: `"Ora Lassila"`

6.8

Beispiel

Einheitliche Darstellung:
(Resource , property, Resource)

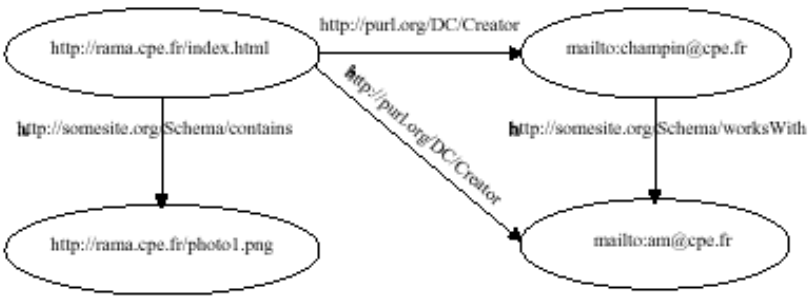
Ein RDF Tripel heißt Aussage (Statement)

Properties sind "first class": können selbst eine URI besitzen

Im Beispiel: Creator ist im bibliographischen Standardvokabular Dublin Core definiert

6.9

- RDF Description als Graph



Mehrwertige Prädikate erlaubt (Creator)

Konkrete Syntax von RDF: XML

6.10

RDF Schema Definition

- RDF Schema:
  - Sprache zur Spezifikation von Schemata
    - Definiert Klassen (class) und deren Eigenschaften (property), die in einer RDF Description mit Werten belegt werden können.
- Analogie zu DB-Schema
  - RDF Schema -Sprache -> Datenmodell (relational, hierarchisch, ...)
  - Konkretes RDF-Schema: legt die Klassen und Eigenschaften einer Anwendung(s-Domäne) fest

6.11

```
<rdf:RDF xml:lang="en"
                 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
                 xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-
   19990303#">

             <rdfs:Class rdf:ID="Person">
               <rdfs:comment>The class of people.</rdfs:comment>
               <rdfs:subClassOf
   rdf:resource="http://www.classtypes.org/useful_classes#Animal"/>
             </rdfs:Class>

             <rdf:Property ID="maritalStatus">
               <rdfs:range rdf:resource="#MaritalStatus"/>
               <rdfs:domain rdf:resource="#Person"/>
             </rdf:Property>

             <rdf:Property ID="ssn">
               <rdfs:comment>Social Security Number</rdfs:comment>
               <rdfs:range
            rdf:resource="http://www.datatypes.org/useful_types#Integer"/>
               <rdfs:domain rdf:resource="#Person"/>
             </rdf:Property>

          ..................
```

```
 <rdf:Property ID="age">
      <rdfs:range
 rdf:resource="http://www.datatypes.org/useful_types#Intege
 r"/>
      <rdfs:domain rdf:resource="#Person"/>
 </rdf:Property>

 <rdfs:Class rdf:ID="MaritalStatus"/>

  <MaritalStatus rdf:ID="Married"/>
  <MaritalStatus rdf:ID="Divorced"/>
  <MaritalStatus rdf:ID="Single"/>
  <MaritalStatus rdf:ID="Widowed"/>

</rdf:RDF>
```

Definition von RDF

- RDF – Konzepte
  - Grundlegende Konzepte:
    - Resource, Property, Statement
  - Schema-Definitionskonzepte zur Definition von neuen Vokabularen
  - Hilfskonzepte zur Vereinfachung der Darstellung
    - Container, Klassen, Typen, ...
- Definiert in

  http://www.w3.org/TR/1999/REC-rdf-syntax-19990222#

  http://www.w3.org/TR/1999/PF-rdf-schema-19990303#

  (Namensräume rdf bzw. rdfs)

6.14

Grundlegende Konzepte

`rdf:Resource`

innerhalb desselben Kontext durch URI identifizierbares Objekt

beachte Kontext:
z.B. W3C-Logo (mit URI !) wird abhängig vom Browser in verschiedenen Bildformaten gelesen
Wetterbericht (mit gleicher URI) liefert täglich neuen Inhalt

`rdf:Property`

Eigenschaft sind Resourcen und können deshalb selbst beschrieben werden.
Können unabhängig definiert werden

6.15

Grundlegende Konzepte

`Rdf: Statement`

Eine Resource, die ein Tripel
`(resource, property, resource)` reifiziert

Reifikation: Vergegenständlichung
Beispiel: Relation R mit Attribut Farbe
Reifikation von Farbe zu Relation F mit Attributen Wert, Spektralfrequenz, ... (etc.)

Erlaubt Aussagen über das reifizierte Ding

6.16

Reifikation

- Beispiel : 'Laut Chris hat "Ora Lassila" das Objekt http://.... gemacht.'

Subject (Resource):

Subjekt (Resource)	http://www.w3.org/Home/Lassila
Prädikat:	Creator
Objekt:	"Ora Lassila"

Prädikat: attributed_to

Object (Resource): http://www.w3.org/people/Chris

6.17

Schema Definition

`rdfs:subPropertyOf`

Beispiel: wenn mother `subPropertyOf` parent, dann soll jede Resource mit Property mother auch property parent haben

Anwendung: Suche nach parent-Wert einer Person auch erfolgreich, wenn Properties mother und/oder father definiert sind

Semantik:

$\forall s, p1,o,p2\ T(s,p1,o) \Rightarrow T(p1,$ `rdfs:subPropertyOf` $,p2)$
$\Rightarrow T(s,p2,o)$

$\forall p1,p2,p3\ T(p1,$ `rdfs:subPropertyOf` $,p2) \Rightarrow$
$T(p2,$ `rdfs:subPropertyOf` $,p3)$
$\Rightarrow T(p1,$ `rdfs:subPropertyOf` $,p3)$

- $T(p,$ `rdfs:subPropertyOf` $,p)$ (nicht in RDF auszudrücken)

6.18

Schema Definition

`rdfs:Class, rdfs:type, rdfs:subClassOF`

Klassen: Mengen gleichartiger Resourcen

Zugehörigkeit einer resource r zu einer Klasse c
durch (r, `rdfs:type`, c)

Semantik:

∀ *x, c1,c2 T(x,* `rdfs:type`*, c1)* ∧ *T(c1,* `rdfs:subClassOf` *,c2)*
⇒ *T(x,* `rdfs:type`*, c2)*

analog zu:
`rdfs:subPropertyOf`: Transitivität, keine Zyklen

6.19

```
<rdf:RDF xml:lang="en"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">
       <rdf:Description ID="registeredTo">
          <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-
   ns#Property"/>
           <rdfs:domain rdf:resource="#MotorVehicle"/>
           <rdfs:range rdf:resource="#Person"/>
       </rdf:Description>

       <rdf:Description ID="rearSeatLegRoom">
          <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-
   ns#Property"/>
            <rdfs:domain rdf:resource="#PassengerVehicle"/>
            <rdfs:domain rdf:resource="#Minivan"/>
            <rdfs:range
   rdf:resource="http://www.w3.org/datatypes#Number"/>
         </rdf:Description>
  </rdf:RDF>
```

Schema Definition

`rdfs:domain, rdfs:range`

Drücken constraints von rdf:property aus:
- Welche Resourcen dürfen die Property haben? (domain)
- Welchen Wert (eindeutig!) darf eine Property haben

Werte in beiden Fällen eine `rdfs:class`

Entspricht einer Typ-Constraint von `rdf:Property` auf Schemaebene. Allgemeiner:

`rdfs:domain, rdfs:range` sind Werte von `rdfs:ConstraintProperty`, die Subklasse von `rdf:Property` ist

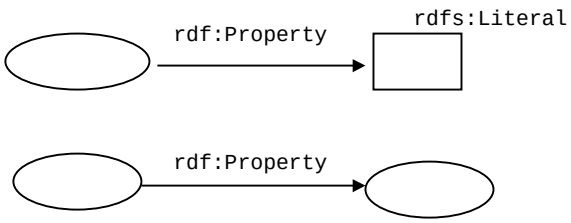
6.21

Grundlagen

`rdfs:Literal`

definiert die Menge der Literale

Degenerierte Klasse, in der Spezifikation nicht als Klasse (also Menge der Objekte, die Literale sind) behandelt.



6.22

Hilfskonzepte

`rdf:Container`

Container sind Kollektionen von Resourcen, Subklassen sind

- `rdf:Bag` ungeordnete Menge
- `rdf:Seq` Sequenz
- `rdf:Alt` Alternative

6.23

Beispiel (Bag): Darstellung des Satzes

The students in course 6.001 are Amy, Tim, John, Mary, and Sue.

```
<rdf:RDF>
   <rdf:Descriptionabout="http://mycollege.edu/courses/6.001">
      <s:students>
        <rdf:Bag>
          <rdf:li resource="http://mycollege.edu/students/Amy"/>
          <rdf:li resource="http://mycollege.edu/students/Tim"/>
          <rdf:li resource="http://mycollege.edu/students/John"/>
          <rdf:li resource="http://mycollege.edu/students/Mary"/>
          <rdf:li resource="http://mycollege.edu/students/Sue"/>
        </rdf:Bag>
      </s:students>
    </rdf:Description>
  </rdf:RDF>
```

6.4 Verwaltung semistrukturierter Daten

- was ist das?
- RDF-Datenbanken: erste Versuche
- Datenverwaltung für XML-Dokumente
  - Daten in XML-Gewand
  - eigentlich XML-Dokumente
  - Abbildung auf relationale und objektrelationale DBS

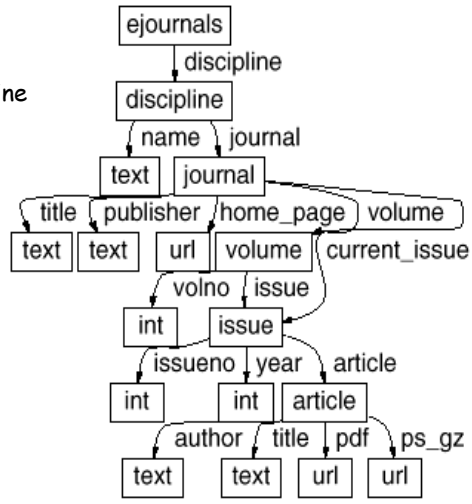
Semistrukturierte Daten

- Semistrukturierte Daten
  - kein festes Schema
  - zeitveränderliche Struktur
  - große Variabilität
  - „Viel Struktur, wenig Daten"

  Beispiel: Web-Daten
    - XML-Dokumente
    - Textliche Dokumente
    - RDF – Beschreibungen
    - .....
- Datenmodell: oft Graphen (bzw. Bäume)
  - Warum „Datenmodell"?
    - Trennung von Struktur und Inhalt erleichtert Anfragen
    - Anfrageverarbeitung wird erleichtert

Semistrukturierte Daten

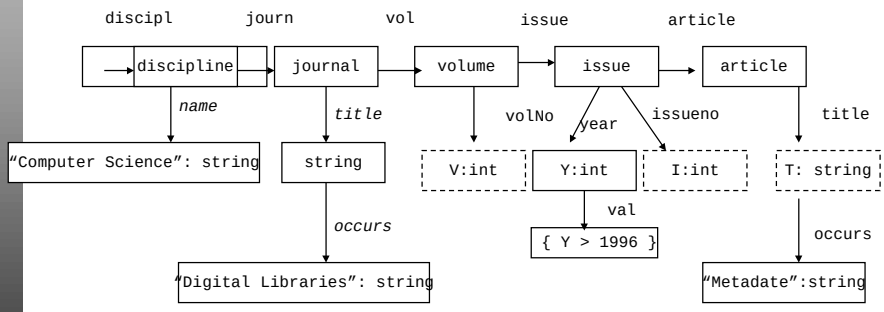
Beispiel:
Struktur der
Web-site f. eine
Zeitschrift



6.27

Semistrukturierte Daten

- Anfragen als Teilgraphen



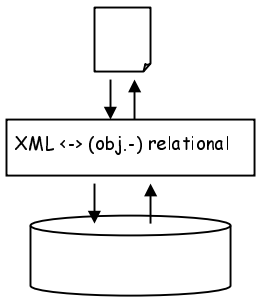
6.28

Semistrukturierte Daten

- Content-Management
  - Verwaltung aller anfallenden elektronischen Informationen (z.B. in einem Unternehmen)
  - Schwerpunkt semistrukturierte Daten
  - Inhaltlich: Web-basierte Daten
    - Bsp.: Internetbroker
      Börsenkurse (HTML / XHTML - Darstellung)
      Presseberichte
      Persönliche Profile .....
    - Bsp. Hochschulverwaltung
      Lehrveranstaltungen, Web-Seiten dafür, Studienpläne, ...
- Datenverwaltung:
  Standard-DBS, Information Retrieval-System oder was??

Semistrukturierte und konventionell Daten

- Derzeitige Schwerpunkte
  - Datenverwaltung von XML-Dokumenten als Spezialfall von Semistrukturierten Daten
  - Verwendung von relationalen / objekt-relationalen DBS
  - Bisher wenig Ansätze für Verwaltung von großen RDF-Datenmengen (keine kommerziellen)
  - Tripelstruktur einfacher als Bäume / Graphen?



RDF-Datenverwaltung

- Was ?
  - RDF-Statements (`Subject, Prädikat, Objekt`)
  - UIDs
  - Literale
  - Namensraum
- Probleme:
  - Skalierbarkeit: 1 G Tripel?
  - Welche Arten von Anfragen?
  - Effizienz!
  - Optimierung: Indexe?
  - Overhead?
  - Datei, Relational, Main-Memory-DB mit Persistenz?

6.31

Mögliches Rel. Schema für RDF*

- Resourcen
- Namespace
- Literale …
- Statement

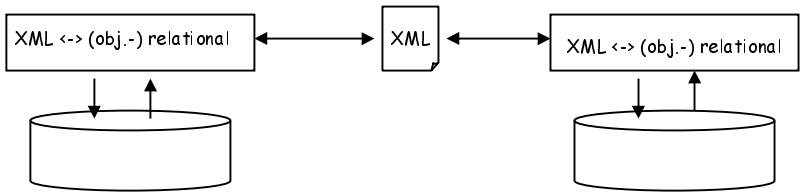
```
CREATE TABLE RDFresource
 (Id INTEGER not null primary key,
  NS INTEGER not null,
  RoName varchar(255))

CREATE TABLE RDFNamespace
 (Id INTEGER not null primary key,
  NsName varchar(255))

CREATE TABLE RDFStatement
 (Id INTEGER not null primary key,
  Subject INTEGER not null,
  Predicate INTEGER not null,
  Object INTEGER,
  Literal INTEGER,
  ResType CHAR(1)) /* literal|resource?
```

Nach B. McBride

6.32

Datenverwaltung für XML

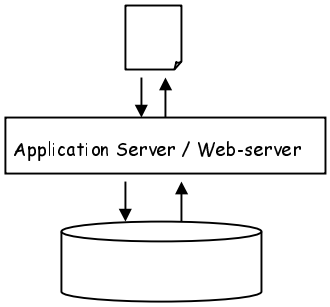
- Warum eigentlich?
  - Szenario 1
    Hersteller X kauft Teile bei Zulieferer Y und verwendet das XML/EDI Protokoll
    (Electronic Data Exchange – Protokoll zur Übermittlung von Handelsdaten, neuerdings in XML-Kodierung)



Datenaustausch: Keine Veranlassung, XML-Dokumente in DBS zu speichern

Datenverwaltung für XML

- Szenario 2
  Internetshop biete Produkte aus einem Datenkatalog mit vielfältigem Angebot an („Otto-Katalog")



Angebot muss abfragbar sein, verschiedenste Kriterien
(Anfragen mit DBS- und IR-Anteilen)
„Fahrrad mit Alurahmen, möglichst rot, höchstens 800,-DM"

Datenverwaltung für XML

- Datenorientiert (data centric documents)

```
<Orders>
   <SalesOrder SONumber="12345">
        <Customer CustNumber="543">
           <CustName>ABC Industries</CustName>
           <Street>123 Main St.</Street>
           <City>Chicago</City>
            ....
        </Customer>
       <Line LineNumber="1">
           <Part PartNumber="123">
              <Description>
                 <p><b>Turkey wrench:</b><br />
                 Stainless steel, one-piece construction,
                 lifetime guarantee.</p>
              </Description>
              <Price>9.95</Price>
           </Part>
           <Quantity>10</Quantity>
        </Line>
        .......
    </SalesOrder>
 </Orders>
```


Datenverwaltung für XML

- Dokument-orientiert (document centric)

```
<Product>
  <Name>Variabler Maulschlüssel</Name>
   <Developer>Full Fabrication Labs, Inc.</Developer>
   <Summary>Großer, verstellbarer Schraubenschlüssel</Summary>
   <Description>
   <Para>Der Engländer besteht aus erstklassigem Stahl und
    besitzt einen gummierten Handgriff. Die Maulgröße liegt
    zwischen 0 und 32 mm.   </Para>
   <Para>Sie könnenPara>

   <List>
   <Item><Link URL="Order.html">Bestellen</Link></Item>
   <Item><Link URL="Wrenches.htm">Andere Werkzeuge ansehen </Link></Item>
   <Item><Link URL="catalog.zip">Den Katalog herunterladen</Link></Item>
   </List>
   <Para> Der Schraubenschlüssel kostet 15.33 Euro inkl. MWSt. Wenn
          Sie jetzt bestellen, erhalten Sie zusätzlich unsere
          wertlose Hobbybastler-Fibel.</Para>
   </Description>
</Product>
```

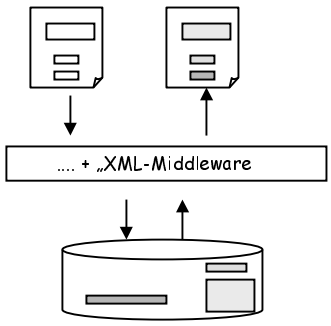
Datenverwaltung für XML

- Probleme
  - Oft keine klarer Unterschied zwischen daten- und dokumentorientiert (Bsp: Preisangabe)
  - XML als Datenrepräsentation zwischen DB
    - Datentypen?
    - Unterschiedliche Dialekte (der DDL)
  - -> für Interoperabilität DB <-> DB gibt es bessere Lösungen
  - Meist Verarbeitungsschritte involviert (-> Szenario 1)
    - Auftrag aus lokaler DB erzeugen (XML-Format)
    - Übertragen
    - Verarbeiten XML Dokument mit Speichern der Auftragsdaten in DB

6.37

Datenverwaltung für XML

- Abbildung DB <-> XML-Dokument: zwei Varianten
  - Templates
    keine vordefinierte Abbildung zwischen Datenbankschema und XML-Dokumentstruktur



6.38

```
<?xml version="1.0"?>
 <FlightInfo>
   <Intro>The following flights have available seats:</Intro>
  <SelectStmt>SELECT Airline, FltNumber, Depart, Arrive FROM
  Flights</SelectStmt>
   <Conclude>We hope one of these meets your needs</Conclude>
  </FlightInfo>
```

Operationen der Mittelschicht:
- Request entgegennehmen
- SQL-Statement ausführen (evtl. indirekt)
- Ergebnisrelation in XML-Format bringen („XML-Middleware")
- Anzeigen

```
?xml version="1.0"?>
  <FlightInfo>
   <Intro>The following flights have available seats:</Intro>
     <Flights>
        <Row>
           <Airline>ACME</Airline>
           <FltNumber>123</FltNumber>
           <Depart>Dec 12, 1998 13:43</Depart>
           <Arrive>Dec 13, 1998 01:21</Arrive>
        </Row>
        ...
     </Flights>
     <Conclude>We hope one of these meets your needs</Conclude>
  </FlightInfo>
```

- Sehr flexibel
- Typisch für Abbildung DB -> XML (auch HTML)-Form, nicht umgekehrt.

Datenverwaltung für XML

- Modell-spezifische Abbildung (relational)

  XML-Dokument-Struktur entspricht dem relationalen Schema (oder Teilen davon) bzw. einer Ergebnismenge.
  - Kanonische Abbildung auf <db>- <table>- <attribut> - Tags

```
<database>
      <table>
         <row>
            <column1>...</column1>
            <column2>...</column2>
            ...
         </row>
         ...
      </table>
      ...
   </database>
```

  - XML / DB-"Middelware" füllt die Elemente oder entnimmt die Daten (wie bei reinem Transfer)

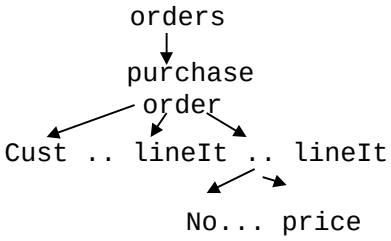
6.41

Datenverwaltung für XML

- Modellspezifische Abbildung (objektrelational oder objektorientiert)
- XML-Dokument lässt sich leichter in objekt-(relationalem) Modell als in (flachem) relationalen Modell darstellen
- Auch hier geht die Ordnung der Teilbäume im XML-dokument verloren: Einbettung
  m: XML–Dok -> ORDBS-Tabellen nicht bijektiv (und damit reversibel)
- Im allgemeinen unkritisch: es kommt auf die Daten an, Darstellung z.B. gemäß XSL-Stylesheet
- Kritisch: Datentypen, XML kennt nur CHAR-Daten
- NULL-Werte: optionale XML-Elemente oder Attribute

6.42

XML-Baumstruktur



Wenn keine hierarchische DB-Struktur: IDREF
Erlaubt XML-Dokumente zu verknüpfen

```
<?xml version=,1.0‘?>
<orders>
  <purchaseorder>
    <Customer>
       <name>  Müller </name>
       ...
    </Customer>
    <lineItem>
      <no> 4711 </>
      ...
    </lineItem>
     <lineItem>
    .....
</orders>
```

Hierarchie in Objektrelationaler DB

```
Create TYPE adressType AS OBJECT
{street varchar2(80},
 city ...};

 Create TYPE lineItemList AS TABLE OF lineItemType;

 Create TYPE lineItemType AS OBJECT
 {no number, name varcchar(40),
 price ...};

 Create TYPE purchaseOrderType AS OBJECT
 {purchaseNo number, customer customerType,
   address addressType, ... lineItemList lineItemListType};

CREATE TABLE purchaseOrderTab AS TABLE OF purchaseOrderType
       NESTED TABLE lineItemList STORE AS...}
```

Objektorientierte Systeme strukturell mindestens so gut geeignet, wie objektrelationale

6.44