Datenbanksysteme

XML und Datenbanken

Burkhardt Renz

Fachbereich MNI
Technische Hochschule Mittelhessen

Sommersemester 2019

Übersicht

- Semistrukturierte Daten
- Datendefinition in XML
  - Dokumenttypdefinition
  - XML-Schema
- Suche in XML
  - Navigation mit XPath
  - Abfragesprache XQuery
- XML und Datenbanken
  - XML aus Datenbank konstruieren
  - XML in Datenbank speichern

Merkmale semistrukturierter Daten

- Daten müssen kein Schema haben
- Struktur ist lokal im Dokument, nicht zentral in einem Katalog – Daten sollen „selbstbeschreibend“ sein
- Wechselnde Struktur, flexibles Schema
- Häufige Änderung der Struktur möglich
- Oft hierarchische Struktur oder Graphen-Struktur

XML – eXtensible Markup Language

- Weiterentwicklung von SGML (Structured Generalized Markup Language)
- Metasprache – Unterschied zu HTML
- Version 1.0 1998, aktuell: Version 1.1 2006
- DTD – Document Type Descriptor
- XML Schema
- XPath – XQuery – XSLT
- Dokumentenzentrierte XML-Dokumente vs. Datenzentrierte XML-Dokumente
- Demo – `wein.xml`

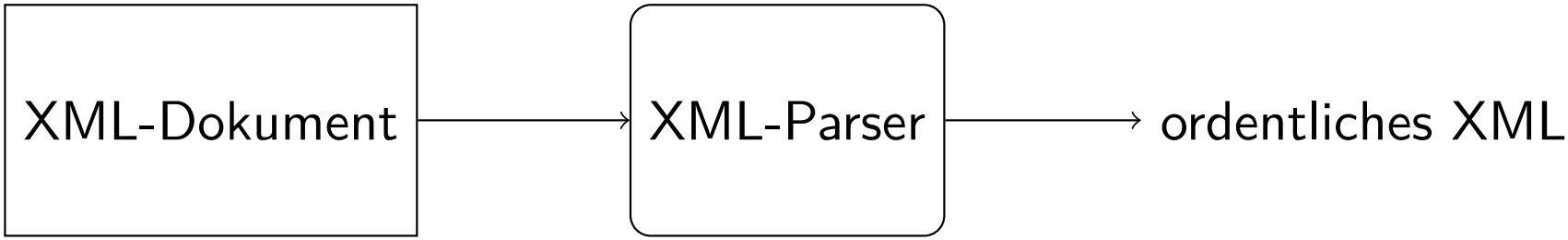
Bausteine von XML

- Geschachtelte *Elemente*, ausgezeichnet durch *Tags*
  `<tag>inhalt</tag>`
- Leere Elemente
  `<tag/>`
- Attribute
  `<tag attribut="wert/>`
- Verarbeitungsanweisungen (*processing instruction*)
  `<?xml version="1.0"?>`
- Kommentare
  `<!-- Kommentar -->`
- Baumstruktur

Regeln für „ordentliche“ XML-Dokumente

- XML-Dokument beginnt mit Deklaration
  `<?xml version="1.0"?>`
- Genau ein Wurzelelement
- Korrespondierende Anfangs- und Ende-Tags
- Namen von Elementen und Attributen beginnen mit einem Buchstaben
- Metazeichen werden maskiert:
  `&lt;` für <,
  `&gt;` für >,
  `&amp;` für &,
  `&quot;` für " und
  `&apos;` für ’.

Verarbeitung von XML



Typen von Parsern

- DOM-Parser (*Document Object Model*)
- SAX-Parser (*Simple API to XML*)

Demo

- `xmllint --noout wein.xml`

Übersicht

- Semistrukturierte Daten
- Datendefinition in XML
  - Dokumenttypdefinition
  - XML-Schema
- Suche in XML
  - Navigation mit XPath
  - Abfragesprache XQuery
- XML und Datenbanken
  - XML aus Datenbank konstruieren
  - XML in Datenbank speichern

DTD (Dokumenttypdefinition)

- Festlegung der Struktur eines Typs von XML-Dokumenten
- Definition ähnlich der Definition einer Grammatik
- Kann Teil des XML-Dokuments sein
- oder in eigener Datei

Beispiele

- Weine mit DTD (standalone) – `wein-dtd.xml`
- Modulbeschreibung mit externer DTD – `cs1020.xml`, `mkintern.dtd`

Demo

- `xmllint --valid --noout wein.xml`

Konzepte von XML-Schema

- Elemente mit Subelementen
- einfache Typdefinitionen basierend auf Basistypen wie string, date etc.
- komplexe Typdefinitionen durch Komposition von Elementen und Attributen
- Erweiterung von Typen
- Restriktion von Typen
- Häufigkeit des Auftretens durch `minOccurs, MaxOccurs`
- ...

Beispiele

```
<xsd:complexType name="Weintyp">
  <xsd:sequence>
    <xsd:element name="Bezeichnung" type="xs:string"/>
    <xsd:element name="Weingut" type="xs:string"/>
    <xsd:element name="Jahrgang" type="xs:year" minOccurs="0"/>
    <xsd:element name="Farbe" type="Weinfarbe"/>
    <xsd:element name="Preis" type="Money"/>
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:simpleType name="Money">
  <xsd:restriction base="xsd:decimal">
    <xsd:fractionDigits value="2"/>
  </xsd:restriction>
</xsd:simpleType>
```

Beispiele und Demo

```
<xsd:simpleType name="Weinfarbe">
  <xsd:restriction base="xs:string">
     <xsd:enumeration value="weiss"/>
     <xsd:enumeration value="rot"/>
     <xsd:enumeration value="rose"/>
  </xsd:restriction>
</xsd:simpleType>
```

Beispiel

- `wein-xsd.xml` mit `wein.xsd`

Demo

- `xmllint --schema wein.xsd --noout wein-xsd.xml`

Übersicht

- Semistrukturierte Daten
- Datendefinition in XML
  - Dokumenttypdefinition
  - XML-Schema
- Suche in XML
  - Navigation mit XPath
  - Abfragesprache XQuery
- XML und Datenbanken
  - XML aus Datenbank konstruieren
  - XML in Datenbank speichern

Navigation mit XPath

- Datenmodell: DOM-Baum
- XPath gibt Kollektionen von Elementen zurück
- XPath = Pfadausdrücke + Bedingungen
- Pfadausdrücke = Adressierung von Dokumentteilen
- Bedingungen = Selektionsprädikate

Pfadausdrücke

- Pfadausdruck = Folge von Navigationsschritten
- Aufbau:
  `axis::node-test[predicate]`
- Beispiel:
  `doc("wein.xml")/descendant-or-self::element(Weingut)`
- kürzer:
  `doc("wein.xml")//Weingut`
- mit Selektionsprädikat:
  `doc("wein.xml")//Wein[Weingut="Louis Max"]`

Achsen für die Navigation

Achse	**Knoten**	**Abk**
child	direkter Subknoten	(leer)
parent	Elternknoten	./..
self	aktueller Knoten	.
ancestor	alle übergeordneten Knoten	
ancestor-or-self		
descendant	alle untergeordneten Knoten	
descendant-or-self		.//
following	alle folgenden Elemente	
following-sibling	alle folgenden Geschwister	
preceding	alle vorherigen Elemente	
preceding-sibling	alle vorherigen Geschwister	
attribute	Attributknoten	@
namespace	Namensraumknoten	

Knotentests

Der Knotentest schränkt die Auswahl der Elemente einer Achse ein:

Form	**Test auf Knoten**	**Kurzform**
node()	alle Knoten	
text()	Textknoten	
attribute()	Attributknoten	@*
element()	Elementknoten	*
child::Wein	Alle Kinder namens "Wein"	Wein
attribute::id	Alle Attribute namens ïd"	@id

Selektionsprädikate

- Bedingung in Klammern `[Weingut="Louis Max"]`
- Vergleichsoperatoren: `<`, `<=`, `=`, `!=` etc.
- Logische Operatoren: `and`, `or`
- Arithmetische Operatoren: `+`, `-`, `*`, `div` etc.
- Funktionsbibliothek („tons of functions“)
  Knotenmengen: `last()`, `position()` etc.
  Strings: `starts-with()`, `contains()` etc.
  Numerische Funktionen: `sum()`, `round()` etc.
- Beispiel:

```
Wein[contains(Weingut,"Louis")
     and Preis < "15.00"]
```

XQuery

- XPath zur Adressierung von XML-Dokument-Teilen
- Konstruktoren zum Erzeugen neuer XML-Elemente
- eingebaute und benutzerdefinierte Funktionen
- Datentypen und Operatoren für diese Datentypen
- bedingte Ausdrücke
- quantifizierte Ausdrücke

FLWOR-Ausdrücke

gesprochen: *flower*

```
for $var in <expr>
let $var := <expr>
where <condition>
order by <expr>
return <expr>
```

alle Klauseln bis auf `return` sind optional
`for` und `let` können mehrfach, auch abwechselnd vorkommen.

Beispiele

```
for $w in doc("wein.xml")//Wein
where $w/Jahrgang = 2004
return $w
```

```
for $w in doc("wein.xml")//Wein
order by $w/Jahrgang
return $w
```

Übersicht

- Semistrukturierte Daten
- Datendefinition in XML
  - Dokumenttypdefinition
  - XML-Schema
- Suche in XML
  - Navigation mit XPath
  - Abfragesprache XQuery
- XML und Datenbanken
  - XML aus Datenbank konstruieren
  - XML in Datenbank speichern

XML-Dokumente aus Datenbank konstruieren

- Erzeugen von XML-Elementen: `xmlelement`
- Erzeugen von Attributen: `xmlattributes`
- Erzeugen von XML-Sequenzen: `xmlforest`
- Aneinanderhängen von XML-Elementen: `xmlconcat`
- Aggregation: `xmlagg`
- Demo wein-xml-out.sql

XML in Datenbanken speichern

- SQL-Datentyp `xml`
- für einzelne XML-Elemente
- für vollständige XML-Dokumente
- Prüfung von XML
- Validierung von XML-Dokumenten
- XPath und XQuery in der Datenbank verwenden
- Demo wein-xml-in.sql