XML-Datenbanken

Autor: Nan Zhang

Veranstaltung: XML in Bioinformatik

Übersicht

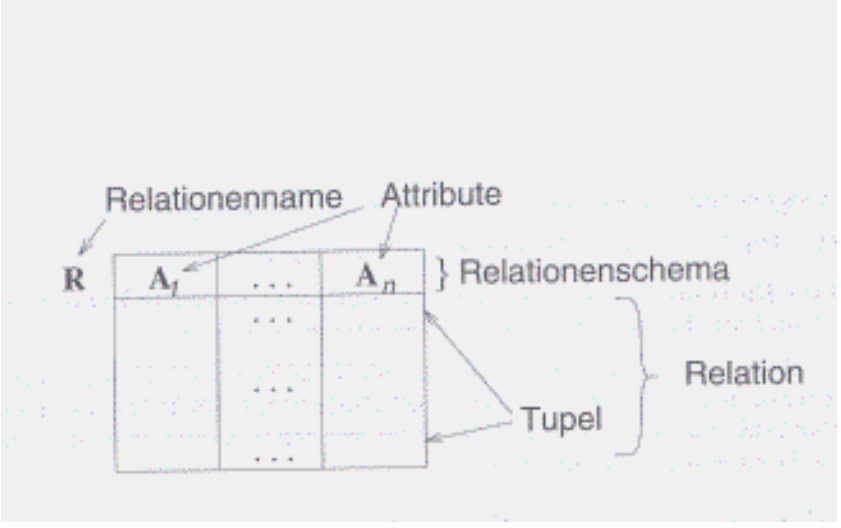
- Relationale Datenbanken
  - Definitionen
  - SQL
- XML-Datenbanken
  - XPath
  - XQuery

Aufbau eines DB-Systems

- Datenmodell
  - Strukturierung der Daten
  - Beispiele: Relationales Datenmodell (Relationen), OO-Datenmodell (Objekte), XML-Datenmodell (Bäume)
- DB-Management-System
  - Verwaltung der Daten
- Anfragesprache
  - Zum Dialog zwischen Applikation und DB-System

Relationale Datenbanken

- Ansammlung von Tabellen
  - Attribut—Spaltenüberschrift
  - Relationenschema—Strukturinformation
  - Tupel—eine einzelne Zeile der Tabelle



Beispiel –Relationale Datenbanken

- Bibliothekausleihe aus zwei Tabellen Ausleihdaten und Bücherinformationen

Ausleih

INVENTARNR	NAME	BENUTZNR
4711	Meyer	150230
1201	Schulz	140160
0007	Müller	150180
4712	Meyer	150230

Buch

INVENTARNR	TITEL	ISBN	AUTOR
0007	Dr.No	3-125	James Bond
1201	Objektbanken	3-111	Heuer
4711	Datenbanken	3-765	Vossen
4712	Datenbanken	3-891	Ullman
4717	XML in Bioinformatik	3-999	Wirth

SQL-eine standardisierte Datenbanksprache

- alle nötigen Sprachbausteine für den Umgang mit einer relationale Datenbank
  - Data Definition Language (DDL)
  - Data Manipulation Language (DML)
  - Data Control Language (DCL)

Data Definition Language (DDL)

- verschiedene Anweisungen, die sich mit der Definition von Daten beschäftigen
  - leere Datenbank einrichten (z.B.Tabellen , Indizes, Views und weitere Objekte)
  - die Strukturdefinitionen existierender Objekte verändern und Objekte löschen.
- Syntax Beispielen in SQL
  - **create database** datenbankname --- Eine Datenbank datenbankname wird erzeugt.
  - **drop database** datenbankname --- Eine Datenbank datenbankname wird gelöscht.
  - **create table** tabellenname (spaltenname datentyp **not null**)
    --- Erzeugt eine neue Tabelle „tabellenname" und eine Spalte „spaltenname". zulässiger Datentyp . Die Option not null -- in jeder Spalte muss ein Wert stehen

Data Manipulation Language (DML)

- Die Manipulation von Daten und deren Auswertung. Operatoren zum Einfügen, Ändern Löschen.
- Anweisung SELECT hat eine zentrale Bedeutung.
- Beispiel
  - **select … from** tabellenname **[where** bedingung**]**
- konkret in unserem vorherigen Beispiel:
  - **select** name **from** Ausleih
  - die Ergebnisrelation

Name
Meyer
Schulz
Müller
Meyer

Data Control Language (DCL)

- In der DCL sind die Anweisungen zur Steuerung der Vergabe von Zugriffsrechten zusammengefasst, mit denen man Benutzern Systemprivilegien und Rechte auf Datenbankobjekte gewähren und entziehen

XML-Datenbanken

- XML bietet viele Sache ,um ein Datenbanksystem zu bauen.

  storage: XML- documents

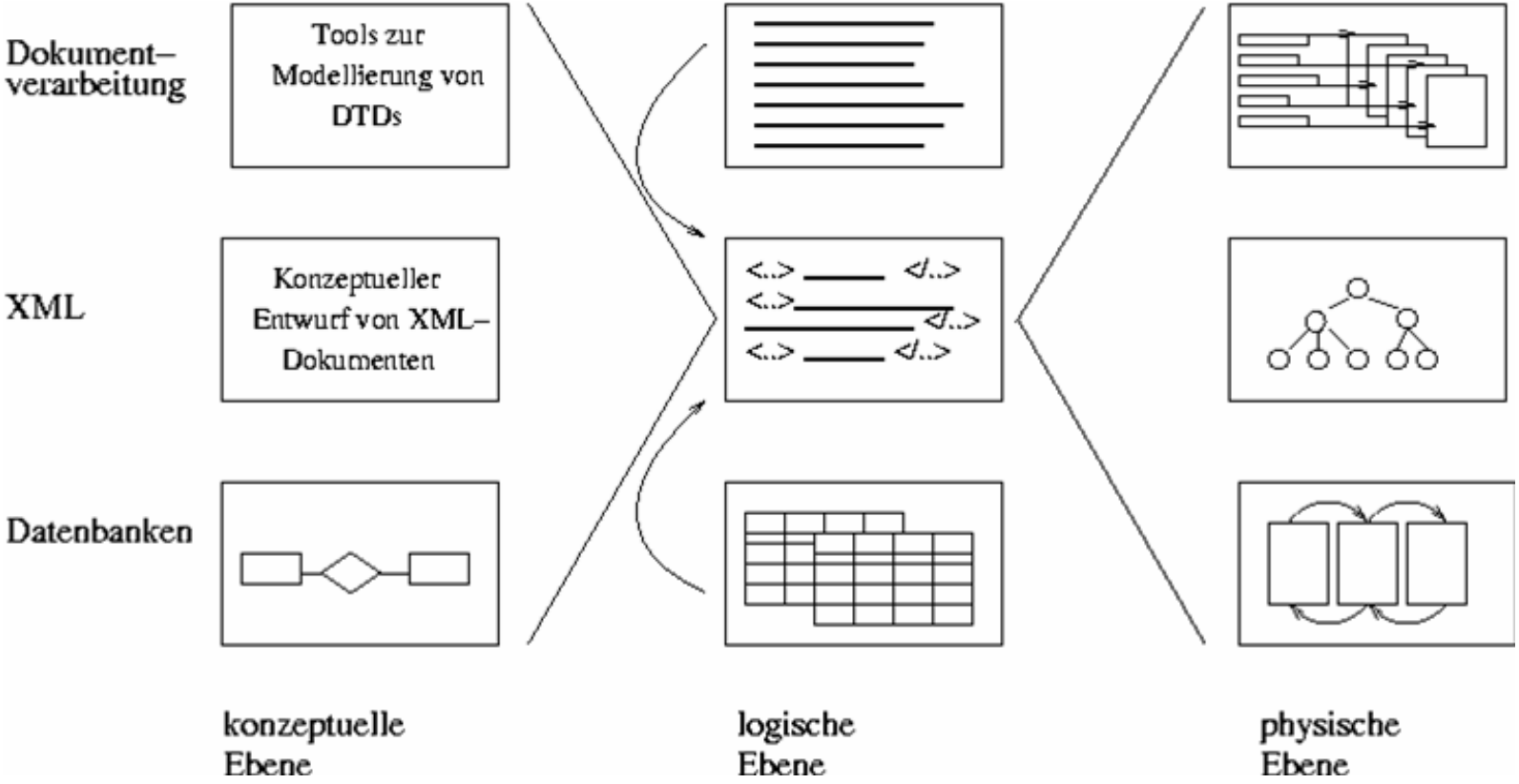
  schemas: DTD , XML schema languages

  query languages XQuery, XPath, XQL,etc.

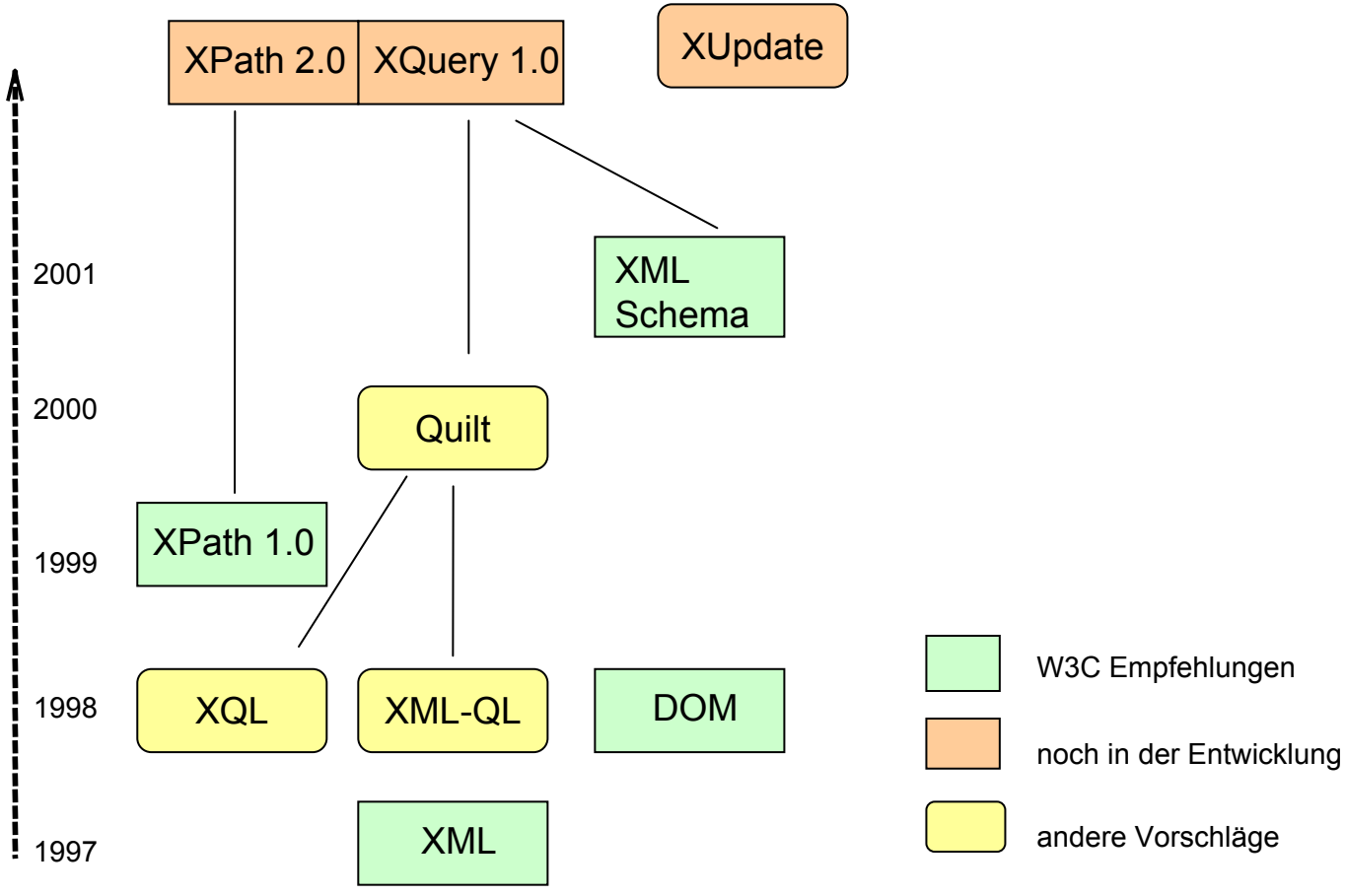
  programming interfaces :SAX, DOM

- Vorteile: Die Benutzer kann passendes Format selbst auswählen. Größtmögliche Flexibilität.

Drei-Ebenen-Architektur



Zeitliche Entwicklung XML



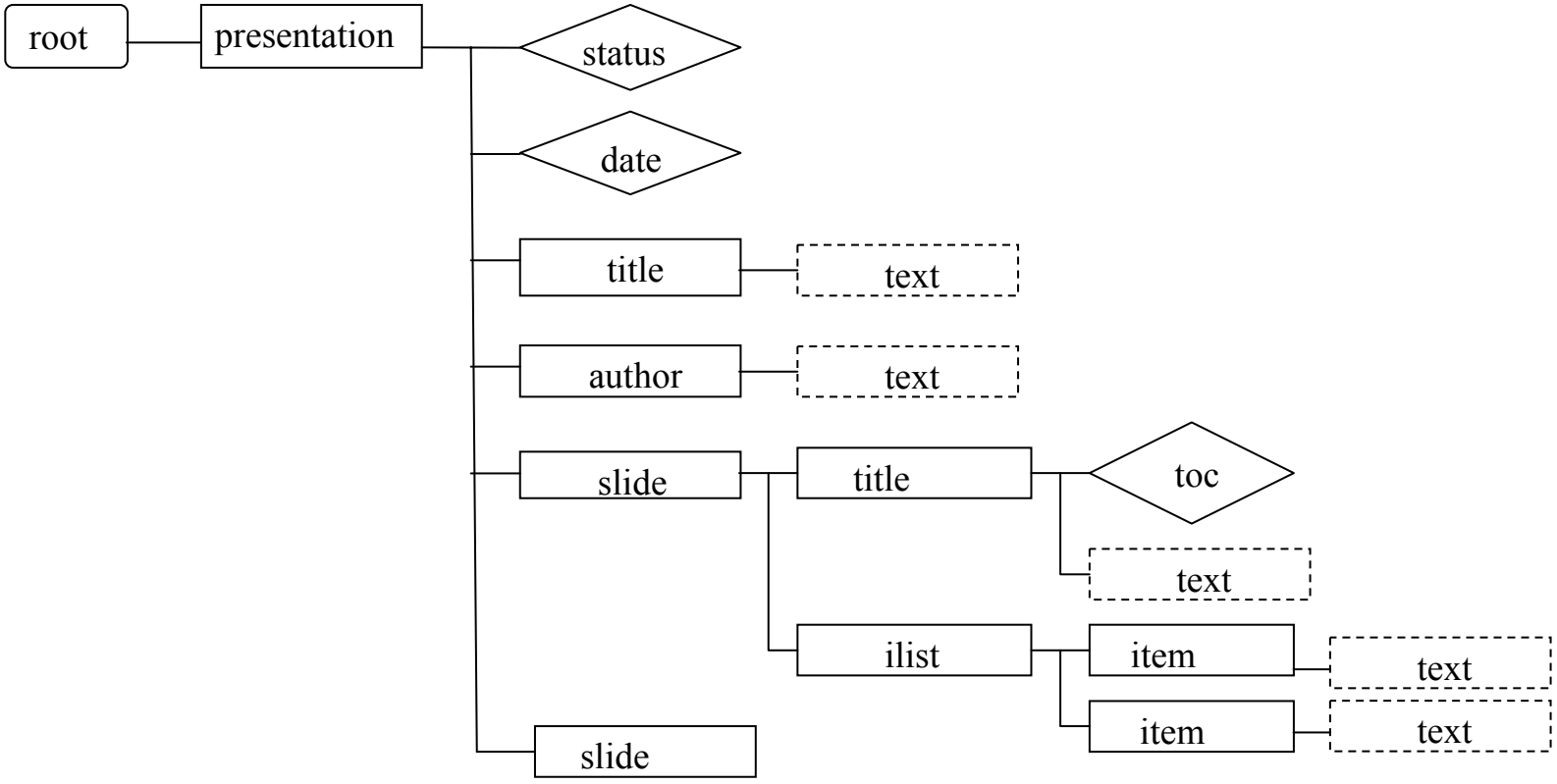
XPath

- Xpath ist eine Sprache, mit der Sie bestimmte Teile eines XML-Dokuments selektieren können.
- XPath verwendet keine XML-Syntax, sondern eine Art Pfadangabe(daher auch der Name XPath) , um Elemente aus einem XML-Dokument anzusteuern
- **Wir betrachten jetzt folgende Ansätzen**
  - **XPath-Datenmodell**
  - **Achsen in XPath**
  - **Knoten auswählen mit XPath**

Ein Beispiel

```
<?xml version="1.0"?>
<presentation status="draft" date="2003-01-29">
   <title>XML</title>
   <author>User</author>
   <slide>
      <title toc="yes">What is XML?</title>
      <ilist>
         <item>XML is not a markeup language (unlike HTML)</item>
         <item>XML instances can be <emph>well formed</emph>or even
          <emph>validating</emph></item>
      </ilist>
   </slide>
  <slide>…</slide>
</presentation>
```

Eine XML-Datei als Baum



XPath-Datenmodell

- XPath kennt folgende Arten von Knoten:
  - Wurzelknoten root
  - Elementknoten
  - Textknoten
  - Attributknoten
  - Namensraumknoten
  - Processing-Instruction-Knoten
  - Kommentarknoten
- Nicht abgebildet sind aber z.B.Dokumenttyp-Deklarationen und CDATA-Abschnitte.

Achsen in XPath

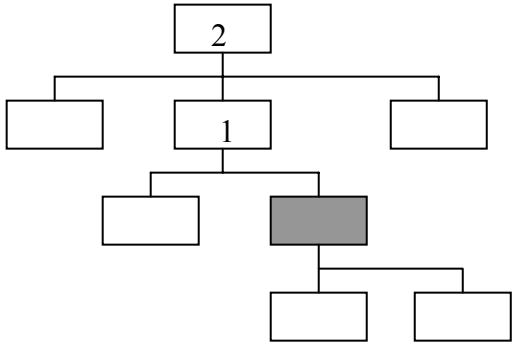
- Achsen sind Wege oder Pfade, entlang derer Sie durch die Baumstruktur navigieren können. beginnen bei Kontextknoten, folgen den Achsen zwischen den Knoten.
- der Kontextknoten in XPath =gegenwärtigen Knoten
- XPath kennt 13 Achsen:

  ancestor, ancestor-or-self, attribute, child, descendant, descendant-or-self, following,following-sibling, namespace, parent, preceding, preceding-sbling, self.

ancestor

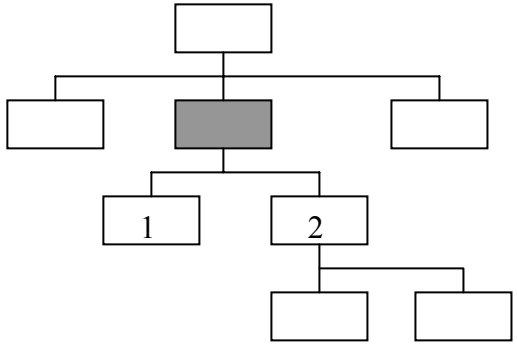
- ancestor: alle Knoten , die Vorfahren des Kontextknotens sind. In umgekehrter Dokumentordnung aufgelistet.

  Der Kontextknoten ist grau markiert.



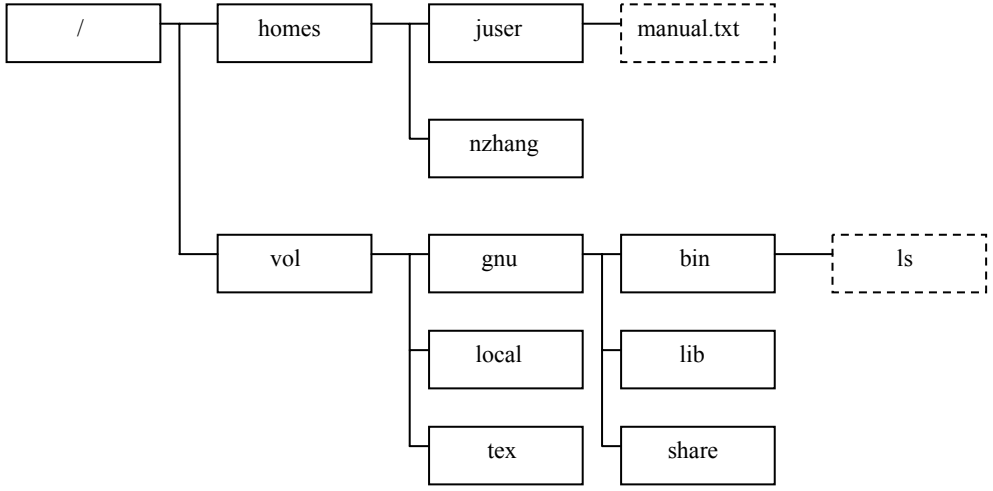
attribute und child

- attribute: Wenn der Kontextknoten ein Element ist, selektiert diese Achse alle seine Attribute in beliebiger Reihenfolge. Ansonsten wird nichts selektiert.
- child: alle Kinder (die direkten Nachfahren) des Kontextknotens in Dokumentordnung aus.



Knoten auswählen mit XPath

- **Lokalisierungspfade:** Es ist sehr ähnlich wie in einem Dateisystem
  - absoluter Pfad : Wegbeschreibung vom Wurzelverzeichnis /vol/gnu/bin/ls
  - relativer Pfad : Wegbeschreibung vom „ aktuellen“ Verzeichnis ../juser/manual.txt
- Unterschied bei XML: gleichnamige Kindknoten erlaubt



Knoten auswählen mit XPath

- **Lokalisierungsschritte**
  Achsenname::Knotentest[Pädikat]
- **Knotentests:** bestimmt den Typ oder den Name des Knotens, der durch den Lokalisierungsschritt ausgwählt werden soll.Knotenfunktionen ,die bestimmte Knotentypen selektieren:
  - * für alle Elementknoten,
  - text() für alle Textknoten,
  - comment() für Kommentarknoten
  - processing-instruction() für Processing-Instruction-Knoten
  - node() für beliebige Knotentypen
- **Prädikate:** Ausdrücke, mit deren Hilfe Sie eine weiter Spezifizierung der Knotenauswahl vornehmen können, also so etwas wie ein zusätzlich Filter.

Beispiele von Anwendung

- **Beispiele:**
  child::tiltle – alle <title> Elemente ,die Kinder des Kontextknoten sind
  attribute:: -- alle Attribute des Kontextknotens*
  /child::presentation/attribute::status -- das status-Attribute von presentation- Element
- **Abgekürzte XPath-Syntax**
  - child:: kann einfach weg lassen
  - // = descendant-or-self::node()
  - . =self::node() ; .. =parent::node()
  - @ =attribute::
- slide/title ; /presentation/author/text()
  /presentation/@status ; /presentation/slide[1]/ilist/item[2]
  /presentation/slide/title[@toc=”yes”]

XQuery

- XQuery wurde erstmals im Februar 2001 als working Draft vorgestellt. Die Sprache übernimmt von XPath die Pfadausdruckssyntax für hierarchische Dokumente. Es ist auch DML.
- **XQuery-Datenmodell**

  ---- eine Erweiterung von XPath

XQuery Ausdrücke

- Anfragen mit verschiedenen Arten von Ausdrücken Formuliert ; Ausdrücke beliebig verschachtelt und zu einem einzigen Anfrageausdruck kombiniert .
- Als Ergebnis liefert ein Ausdruck immer eine Liste, die sowohl einzelne Werte als auch Knoten enthalten kann.

Einfache Beispiele mit XQuery

- Erste Beispiele:
  let $x:=5 let $y:=6 return 10*$x+$y
  > 56
- Sequenz:
  let $a:=3,4
  let $b:=($a, $a)
  let $c:=99
  let $d= ()
  return (count($a), count($b), count($c), count($d))
  >(2, 4, 1, 0)
- Kommentar beginnt mit #

Pfadausdrücke

- gleich wie in XPath 2.0
- Erweiterung gegenüber XPath1.0: ein Ausdruck in XQuery liefert immer eine Sequenz , hat Ordnung (XPath kennt nur Knotenmengen)
- Beispiele:
  - *let $book := document("mybook.xml")/book*
    return $book/chapter

    document Funktion gibt die Wurzelknoten von einem Dokument zurück. Der /book Ausdruck selektiert das <book>-Element direkt unterhalb des Wurzelknotens. Das Programm gibt einen Sequenz aus alle <chapter>-Elemente unter <book>-Element.

  - *return $book//para[@class="waring"]*

FLWR-Ausdrücke

- Die wichtigste Ausdrücke in XQuery -- FLWR-Ausdrücke (gesprochen „Flower-Ausdrücke“) . FOR bzw. LET, WHERE und RETURN .
- *for $ x in (1 to 3) return ($x,10+$x)*
  >1,11,2,12,3,13
- *for $c in customers*
  for $o in orders
  *where $c.cust_id=$*o.cust_ *id and $o.part_id=”xx”*
  return $c.name
- entspricht ähnlich Funktion in SQL (jion)
  select customers.name
  from customers, order
  where customer.cust_id=orders.cust_id
  and orders.part_id=”xx”

Funktionen in XQuery

- *define funktion descendant-or-self($x)*

```
{
   $x,
   for $y in  children($x)
        return descendant-or-self($y)
}
```

descendant-or-self(<a>XY)
> <a>XY; "X"; Y; "Y"

- **Sortieren**
  *$b*ooks sortby (author/name)
- **Was kann XQuery nicht ?**
  Insert ,Delet ,Update

XUpdate

- Erzeugen und Löschen von allen XPath-Elementtypen (element, attribute, text)
- Aktualisierung von Elementen (Update)
- Umbenennen von Elementen

Drei verschiedene Typen von Dokumenten

Daten-zentrierte Dokumente

(strukturiert, regulär
Beispiele: Produktkataloge, Bestellungen, Rechnungen)

Dokument-zentrierte Dokumente

(unstrukturiert, irregulär
Beispiele: wissenschaftliche Artikel, Bücher, E-Mails, Webseiten)

Semistrukturierte Dokumente

(datenzentrierte und dokumentenzentrierte Anteile
Beispiele: Veröffentlichungen, Amazon)

```
<order>
  <customer>Meyer</customer>
  <position>
    <isbn>1-234-56789-0</isbn>
    <number>2</number>
    <price currency=„Euro“>30.00</price>
  </position>
</order>
```

```
<content>
XML builds on the principles of two existing
 languages, <emph>HTML</emph> and
<emph>SGML</emph> to create a simple
mechanism ..
The generalized markup concept ..
</content>
```

```
<book>
  <author>Neil Bradley</author>
  <title>XML companion</title>
  <isbn>1-234-56789-0</isbn>
  <content>
    XML builds on the principles of two existing
    languages, <emph>HTML</emph> and ..
  </content>
</book>
```

- XML-fähige Datenbanken:
  - **Oracle8*i*** RDBMS mit vielen Erweiterungen und Werkzeugen für XML
- Native XML-Datenbank
  - Tamino

Literatur

- Heuer / Saake
  Datenbanken – Konzepte & Sprachen
- Franz-Josef Herpers / Thomas J. Sebestyen
  Das Einsteigerseminar XSL