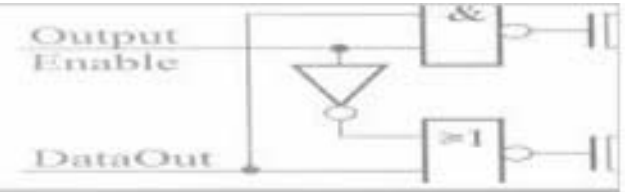


XML - XQuery



Sebastian Krug

XQuery

XML - XQuery

➔ Übersicht

[Einführung](#) | [Anforderungen](#) | [Operationen](#) | [zum Ausprobieren](#)

Sebastian Krug

- XQuery – Einführung
- Anforderungen
- Operationen
- zum Ausprobieren

XQuery: XML Query Language

Standardisierungs-Prozess beim W3C seit Oktober 1999.

Allgemeines Werkzeug für Abfragen von Datenbeständen.

<bib>

<book year="1994">

<title>TCP/IP Illustrated</title>

<author><last>Stevens</last><first>W.</first></author>

<publisher>Addison-Wesley</publisher>

<price>65.95</price>

</book>

<book year="2000">

<title>Data on the Web</title>

<author><last>Abiteboul</last><first>Serge</first></author>

<author><last>Buneman</last><first>Peter</first></author>

<author><last>Suciu</last><first>Dan</first></author>

<publisher>Morgan Kaufmann Publishers</publisher>

<price>39.95</price>

</book>

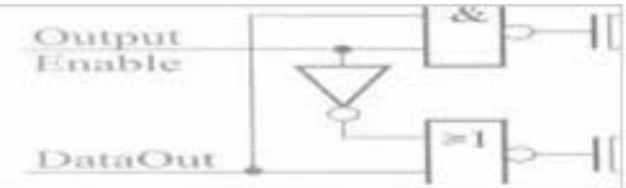
</bib>

Year	Title	Author	Publisher

Year	Price

```
Select * from bib
where publisher= 'Addison-Wesley'
```

```
<bib>
{
  for $b in doc("http://bstore1.example.com/bib.xml")/bib/book
  where $b/publisher = "Addison-Wesley"
  return $b
}
</bib>
```



- Ad-hoc-Formulierung Anfragen können vom Nutzer direkt gestellt werden.
- Deskriptivität Nutzer umschreibt das gewünschte Ergebnis.
- Kompaktheit
- Mengenorientiertheit
- Adäquatheit
- Orthogonalität
- Abgeschlossenheit Resultat kann wieder verwendet werden.
- Vollständigkeit verlustfreie Wiedergewinnung aller Daten.
- Optimierbarkeit
- Effizienz
- Sicherheit
- Eingeschränktheit

XML - XQuery

➔ spezielle XML-Anforderungen

Einführung | [Anforderungen](#) | Operationen | zum Ausprobieren

Sebastian Krug

- Einbettung
- Server-Verarbeitung
- Ordnungserhaltung
- Hyperlinks
- Flexible Typen
 - Namensräume
- „robuste“ Fehlerbehandlung



Auswahl eines Dokumentes, oder einzelner Elemente anhand des Inhaltes, der Struktur oder der Attribute.

Werteselektion

```
<bib>
```

```
{
```

```
  for $b in doc("http://bstore1.example.com/bib.xml")/bib/book  
  where $b/publisher = "Addison-Wesley" and $b/@year > 1991  
  return $b
```

```
}
```

```
</bib>
```

Eine Achse und ein Auswahl-Kriterium, getrennt durch 2 Doppelpunkte.
List the descriptions of all items offered for sale by Smith.

```
document("items.xml")/child::*  
  /child::item[child::seller = "Smith"]  
  /child::description
```

„Child-Achse“ ist „default“, wird in der Regel ausgelassen:

```
document("items.xml")  
/* /item[seller = "Smith"] /description
```

Doppel-Slash: über alle Ebenen

List all description elements found in the document items.xml.

```
document("items.xml")//description
```

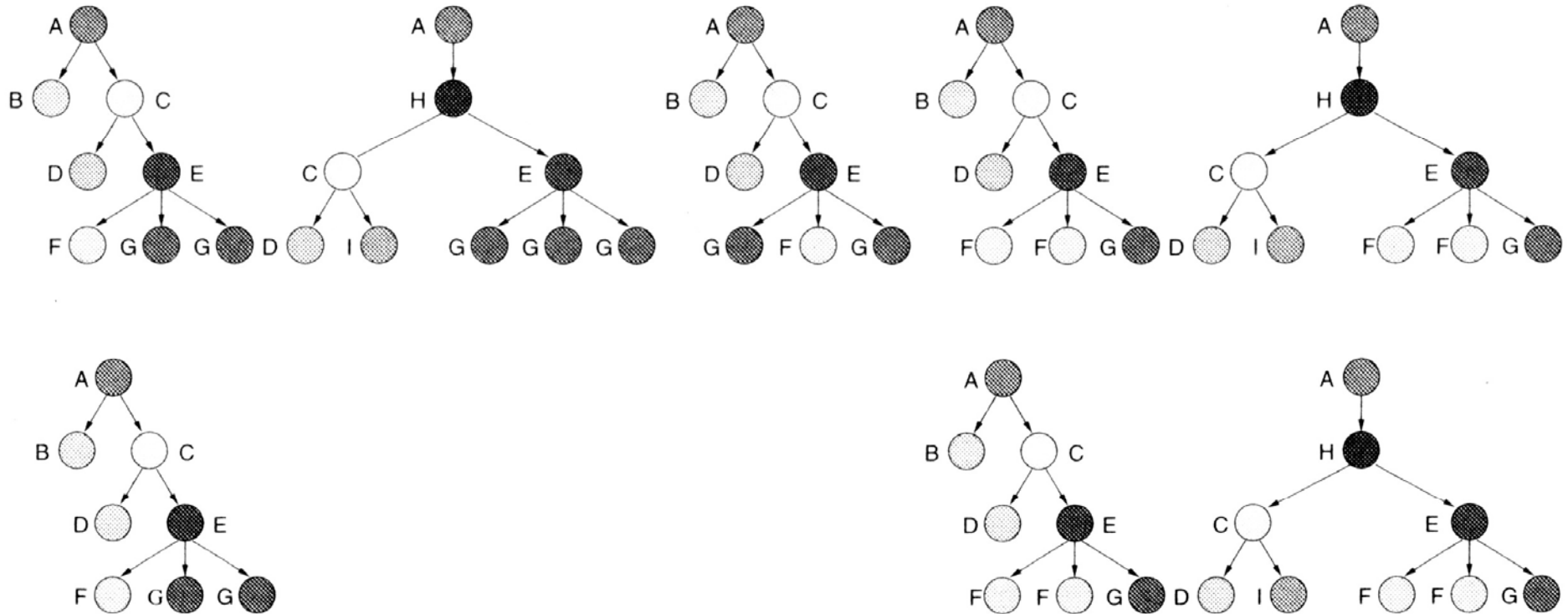
Attribute: @

Find the status attribute of the item that is the parent of a given description.

```
$description/../@status
```



Strukturselektion und Existenz- und Allquantor



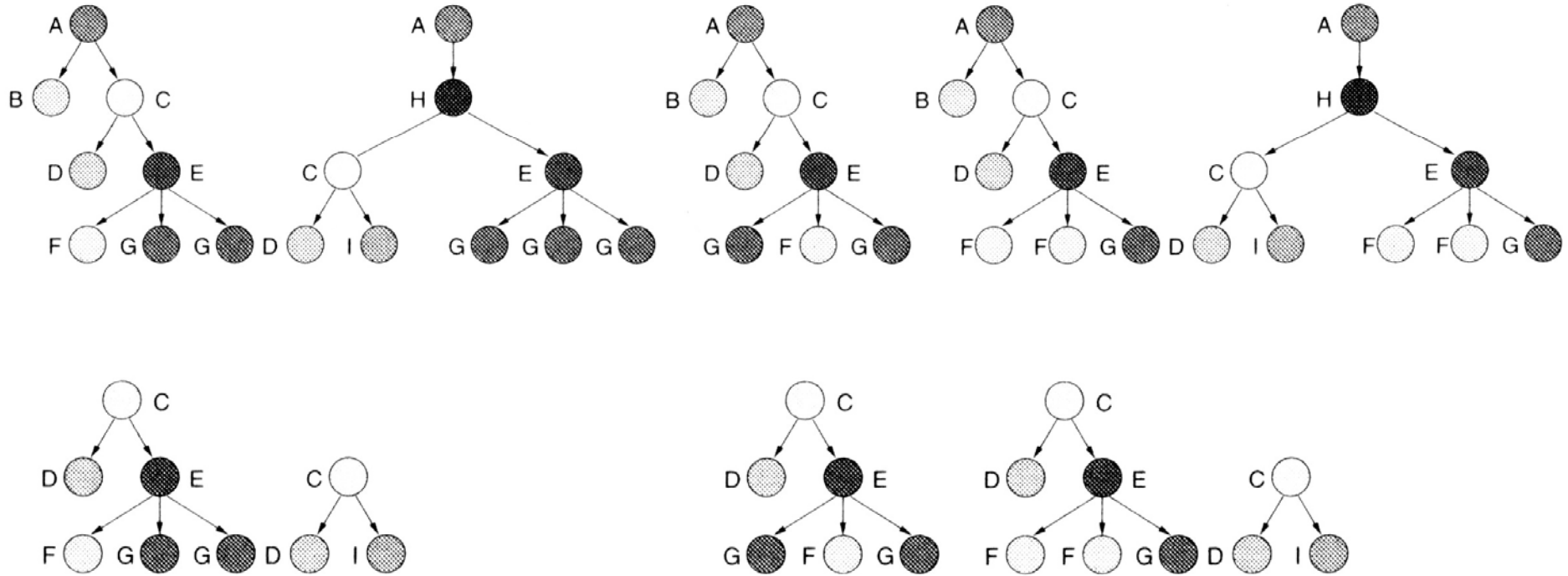
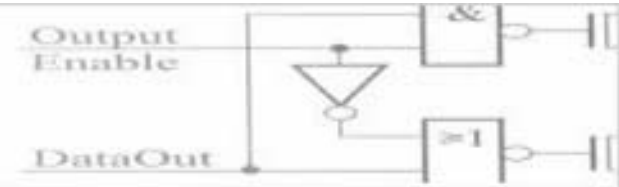
some \$n in (5, 7, 9, 11) satisfies \$n > 10
 every \$n in (5, 7, 9, 11) satisfies \$n > 10

XML - XQuery

➔ Extraktion / Reduktion

Einführung | Anforderungen | **Operationen** | zum Ausprobieren

Sebastian Krug



XML - XQuery

➔ Extraktion / Reduktion

Einführung | Anforderungen | **Operationen** | zum Ausprobieren

Sebastian Krug

<bib>

{

```
for $b in doc("http://bstore1.example.com/bib.xml")/bib/book
where $b/publisher = "Addison-Wesley" and $b/@year > 1991
return $b
```

}

</bib>

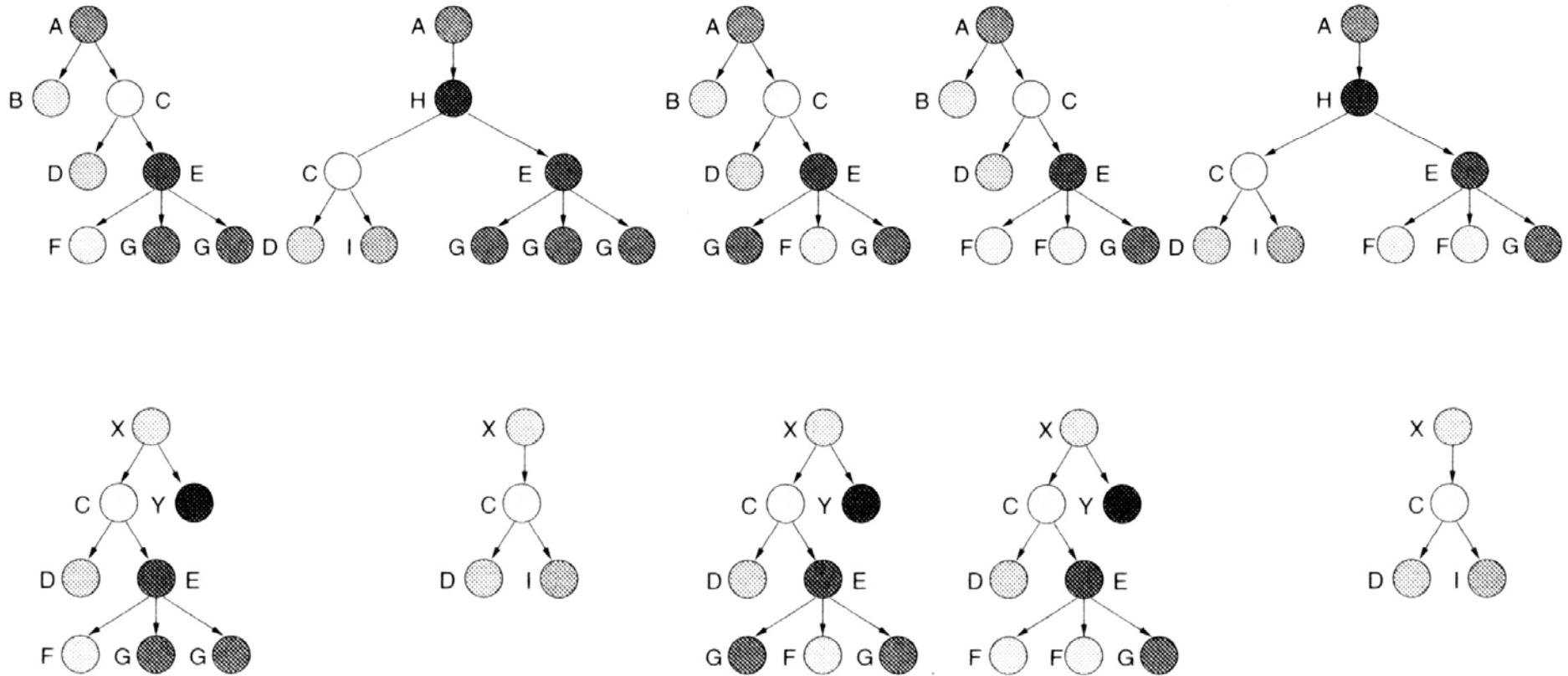
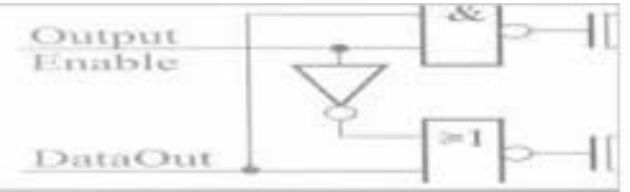
```
for $i in document("items.xml")/*/*item
let $b := document("bids.xml")
    /*/*bid[itemno = $i/itemno]
where count ($b) > 10
return
  <popular-item>
  {
    $i/itemno,
    $i/description,
    <bid-count> {count ($b)} </bid-count>
  }
</popular-item>
```

XML - XQuery

➔ Restrukturierung

Einführung | Anforderungen | **Operationen** | zum Ausprobieren

Sebastian Krug



```
<bib>
{
  for $b in doc("http://bstore1.example.com/bib.xml")/bib/book
  where $b/publisher = "Addison-Wesley" and $b/@year > 1991
  return
    <book year="{ $b/@year }"> { $b/title }
    </book>
}
</bib>
```

Variablen-Vergleichs-Operatoren:

eq, ne, lt, le, gt, ge

item[reserve-price gt 1000]

Generelle Vergleichs-Operatoren:

=, !=, >, >=, <, <=

item[reserve-price > 1000]

Arithmetische Operationen:

+, -, *, div, mod

Vorsicht bei leeren Sequenzen !

Aggregations-Funktionen:

sum, avg, count, max, min

XML - XQuery

➔ weitere Operationen (2)

Einführung | Anforderungen | **Operationen** | zum Ausprobieren

Sebastian Krug

Logische Operatoren:

and, or

[seller eq "Smith" and reserve-price].

Node-Vergleich:

is, isnot

Positions-Vergleich:

$\$node1 \ll \$node2$

```
for $i in document("items.xml")/*/*item
let $b := document("bids.xml")
      /*/*bid[itemno = $i/itemno]
where count ($b) > 10
return
  <popular-item>
  {
    $i/itemno,
    $i/description,
    <bid-count> {count ($b)} </bid-count>
  }
</popular-item>

sortby bid-count descending
```

```
FOR $h IN document („hotels.xml“)//hotel
LET $z := document („zimmer.xml“)//zimmer[hotel=$h/name]
WHERE count ($z) > 10
RETURN
  <hotel> {
    $h/name,
    FOR $zt IN $h/zimmertyp
    LET $ztz := $z/[typ=$zt/typ]
    RETURN
      <zimmerzahl> {
        $zt/typ, <anzahl>{count($ztz)} </anzahl>
      } </zimmerzahl>
  } </hotel>
```

```
<highbid status = "pending">  
  <itemno>4871</itemno>  
  <bid-amount>250.00</bid-amount>  
</highbid>
```

```
<highbid status = "{$s}">  
  <itemno> {$i} </itemno>  
  <bid-amount>  
    { max($bids[itemno = $i]/bid-amount)}  
  </bid-amount>  
</highbid>
```

```
<highbid>  
  {  
    $b/@status,  
    $b/itemno,  
    $b/bid-amount  
  }  
</highbid>
```

element

```
element {name($e)}  
  {$e/@*, data($e)*2}
```

attribute

```
attribute  
  {if $p/sex="M" then "father" else "mother"}  
  {$p/name}
```

➔ define function (1)

Einführung | Anforderungen | **Operationen** | zum Ausprobieren

Sebastian Krug

```
define function
  highbid(element $item)
  returns decimal
  {
    max(document("bids.xml")
      //bid[itemno = $item/itemno]/bid-amount)
  }
```

```
highbid(document("items.xml")
  //item[itemno = "1234"])
```

Keine Überladung.

➔ define function (2)

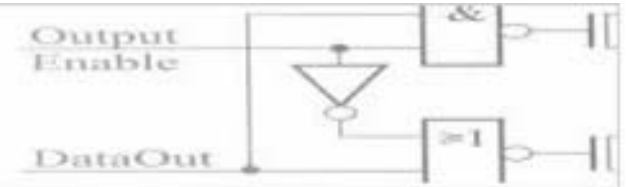
```
define function defaulted
  (element? $e, anySimpleType $d)
  returns anySimpleType
  {
    if (empty($e)) then $d
    else if (empty($e/*)) then $d
    else data($e)
  }
```

```
for $e in $emps
return
  <emp>
  {
    $e/name,
    <pay> { $e/salary
      + defaulted ($e/commission, 0)
      + defaulted ($e/bonus, 0) }
    </pay>
  }
</emp>
orderby(pay)
```

Rekursion möglich.

XML - XQuery

➔ Kombination von Nodes



Einführung | Anforderungen | **Operationen** | zum Ausprobieren

Sebastian Krug

union

intersect

except

Intersect und Except bei 2 verschiedenen Dokumenten nicht sinnvoll, da keine Möglichkeit der selben Node-Identiy.

XML - XQuery

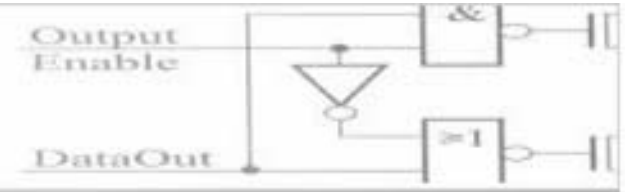
➔ Fragen ?

Einführung | Anforderungen | Operationen | zum Ausprobieren

Sebastian Krug

Fragen ?

XML - XQuery



Einführung | Anforderungen | Operationen | zum Ausprobieren

Sebastian Krug

Schönen Dank.

- IBM Systems Journal, Vol. 41, No. 4, 2002
<http://www.research.ibm.com/journal/sj/414/chamberlin.html> [11.11.05]
- W3C XML Query Use Cases, Working Draft 15, Sep. 2005
<http://www.w3.org/TR/xquery-use-cases/> [11.11.05]
- Klettke, Meyer: XML & Datenbanken; dpunkt.verlag, 2003, Heidelberg
ISBN: 3-89864-148-1

```
for $n in (2, 3)
return $n + 1
```

```
for $m in (2, 3), $n in (5, 10)
return
  <fact>{$m} times {$n} is  {$m * $n}</fact>
```

```
<fact>2 times 5 is 10</fact>
<fact>2 times 10 is 20</fact>
<fact>3 times 5 is 15</fact>
<fact>3 times 10 is 30</fact>
```

XQuery-Webformular:

<http://ingo.fargonauten.de/2005/xml-isy/xquery.html>

XML-File unter:

<http://datenkeller.de/xml-isy/bib.xml>

Aufgabe 1: Liste alle Bücher, die von „Addison-Wesley“ nach 1991 veröffentlicht wurden. Gib dabei Titel und Jahr an.

Aufgabe 2: Für jedes Buch, das wenigstens einen angegebenen Autor hat, liste den Titel und die max. ersten 2 Autoren. Falls das Buch weitere Autoren hat, soll dafür ein leerer „<etc./>“-Tag geschrieben werden.

XML-File unter:

<http://datenkeller.de/xml-isy/auktion.xml>

Aufgabe 3: Führe einen Join auf dem Dokument Auktion aus, die alle „Items“ mit den „Bids“ verbindet und gib die Item-Nr. aus.