

# XML und Datenbanken

## — 10. Übungsblatt: XQuery —

### Allgemeine Aufgabe

Notieren Sie sich eventuelle Verständnisfragen, so dass wir diese im nächsten Online-Treffen klären können. Da es erfahrungsgemäß sehr sill ist: Nehmen Sie sich bitte die Zeit und denken Sie bewusst über mögliche Fragen nach (wenigstens eine). Ich kann auch Studierende drannehmen, die sich nicht gemeldet haben. Ein Mal würde ich wohl verstehen, dass der Stoff so einfach war, dass Sie einfach keine Fragen haben. Wenn sich das wiederholt, müsste ich dann aber umgekehrt prüfungsähnliche Fragen stellen.

### Hausaufgabe

Geben Sie die folgenden Aufgaben bis Montag, 16.01.2023, 16<sup>00</sup>, über die Übungsplattform in StudIP ab. Schreiben Sie die Lösungen in eine `.txt`- oder `.xq`-Datei. Es gibt 3 Punkte pro Anfrage.

Die Abgaben nur stichprobenartig kontrolliert. Wenn Ihre Abgabe nicht kontrolliert wurde, bekommen Sie die volle Punktzahl. Wenn Sie später wegen Plagiaten auffallen, oder bei einer Stichprobe eine fast gar nicht gelöste Aufgabe entdeckt wird, können auch alte Abgaben kontrolliert werden. Dann können auch rückwirkend Punkte abgezogen werden.

Sie benötigen 67% der Hausaufgabenpunkte und eine aktive Mitarbeit in den Übungen für die Studienleistung.

Die „Wiederholungsaufgaben“, also Teil d) bis h), sind nicht abzugeben. Beschäftigen Sie sich aber bitte auch mit diesen Aufgaben. Sie müssen damit rechnen, dass Sie beim Online-Treffen gebeten werden, einen Teil des Vorlesungs-Stoffes zu wiederholen und insbesondere eine der Wiederholungsfragen zu beantworten.

Verwenden Sie nochmals die XML-Datensammlung für klassische Musik-CDs (diesmal aber die deutsche Version):

- Daten:  
[<http://www.informatik.uni-halle.de/~brass/xml18/examples/cd/cd.xml>]
- XML Schema:  
[<http://www.informatik.uni-halle.de/~brass/xml18/examples/cd/cd.xsd>]
- DTD:  
[<http://www.informatik.uni-halle.de/~brass/xml18/examples/cd/cd.dtd>]

Das Dokument enthält die folgenden Elemente:

- CDDB: (komponisten, cds?, solisten?)
- komponisten: (komponist\*)
- komponist: (stueck\*)  
Attribute: knr, vorname, name, geboren, gestorben.
- stueck: (aufnahme\*)  
Attribute: snr, titel, tonart, opus.
- aufnahme: Leerer Inhalt.  
Attribute: aufnnr, orchester, leitung.
- cds: (cd\*).
- cd: (CDAufnahme\*)  
Attribute: cdnr, name, hersteller, anz-cds, gesamtspielzeit.
- CDAufnahme: Leerer Inhalt.  
Attribute: aufnnr.
- solisten: (solist\*).
- solist: (solist\_aufnahme\*)  
Attribute: name.
- solist\_aufnahme: Leerer Inhalt.  
Attribute: aufnnr, instrument.

Schreiben Sie die folgenden Anfragen in XQuery und testen Sie Ihre Lösung mit einer XQuery-Implementierung.

- a) Erzeugen Sie ein XML-Dokument, das alle Komponisten enthält, aber nur mit den Attributen `vorname`, `name`, `geboren`. Die `stueck`-Elemente, die in den `komponist`-Elementen geschachtelt sind, sollen nicht in der Ausgabe erscheinen. Die Ausgabe soll also so aussehen:

```
<HA10>
  <komponist vorname='Georg Friedrich' name='Händel'
    geboren='1685' />
  <komponist vorname='Serge' name='Prokofiev'
    geboren='1891' />
  ...
</HA10>
```

Die Zeilenumbrüche und die Einrückung werden von der Implementierung gewählt und können anders aussehen, als hier gezeigt. Die Reihenfolge der “`komponist`”-Elemente soll die gleiche Reihenfolge wie in der Eingabedatei sein.

- b) Erzeugen Sie eine Liste aller Komponisten, die vor 1700 geboren wurden, geordnet nach dem Geburtsjahr (absteigend). Die Ausgabe soll eine ungeordnete Liste in HTML sein, und wie folgt strukturiert:

```
<ul>
  <li>Lean-Marie Leclair: 1697</li>
  <li>Pietro Locatelli: 1695</li>
  <li>Benedetto Marcello: 1686</li>
  <li>Georg Friedrich Händel: 1685</li>
  <li>Johann Sebastian Bach: 1685</li>
  <li>Domenico Scarlatti: 1685</li>
  <li>Georg Philipp Telemann: 1681</li>
  ...
</ul>
```

- c) Gesucht sind alle Komponisten zusammen mit der Anzahl Stücke des jeweiligen Komponisten. Es sollen nur Komponisten mit mindestens 5 Stücken berücksichtigt werden. Die Ausgabe soll nach der Anzahl Stücke sortiert werden (größte Anzahl zuerst). Die Ausgabe soll als HTML Tabelle geschehen, also folgendermaßen aussehen:

```
<table>
  <tr><th>Name</th><th>Vorname</th><th>Anzahl</th></tr>
  <tr><td>Händel</td><td>Georg Friedrich</td><td>25</td></tr>
  ...
</table>
```

## Wiederholungsaufgaben

Beschäftigen Sie sich mit diesen Aufgaben. Sie brauchen aber nichts abzugeben.

- d) Was würden Sie in einer mündlichen Prüfung auf folgende Fragen zu XPath antworten?
- Nennen Sie einige Unterschiede zwischen XPath 1.0 und XPath 2.0. D.h. welche XPath 2.0 Konstrukte können nicht in einem XSLT Stylesheet verwendet werden, das auch in Browsern funktionieren soll, die nur XPath 1.0 unterstützen? (Das ist noch der Stand der Technik.)
  - Was ist die Bedeutung der folgenden Bedingung in XPath, d.h. für welche Attributwerte ist sie wahr?

```
@weekday = ('Sat', 'Sun')
```

- Warum könnte der Ausdruck “\$x idiv 1” nützlich sein? (Bedenken Sie, dass das Ergebnis von idiv immer eine ganze Zahl ist, während die Argumente von beliebigem numerischen Typ sein können.)

- Was ist der Unterschied zwischen den beiden folgenden XPath Ausdrücken?
  - `if @GUEST_STUDENT = true() then ... else ...`
  - `if @GUEST_STUDENT then ... else ...`
- Und was ist der Unterschied zwischen diesen XPath Ausdrücken?
  - `true`
  - `true()`
- Gibt es einen Unterschied zwischen diesen XPath Ausdrücken?
  - `not /A`
  - `not(/A)`
- Was ist die Bedeutung von “`not(*)`”?
- Wie können Sie prüfen, ob der Wert des Attributes **A** des Kontext-Knotens eine gültige lexikalische Repräsentation einer ganzen Zahl (`xs:integer`) ist (d.h. eine Folge von Ziffern, ggf. mit Vorzeichen)? Schreiben Sie einen XPath Ausdruck, der in diesem Fall den Zahlwert liefert, und sonst die Zahl `-1`.
- Was ist der Unterschied zwischen den beiden folgenden XPath-Bedingungen?
  - `@A castable as xs:positiveInteger`
  - `@A instance of xs:positiveInteger`
- Und was ist der Unterschied zwischen diesen XPath-Ausdrücken?
  - `@A cast as xs:positiveInteger`
  - `@A treat as xs:positiveInteger`
- Was sind die Vor- und Nachteile von statischer Typprüfung in XPath?
- Nennen Sie zehn wichtige XPath-Funktionen. Wie würden Sie ein Handbuch über XPath-Funktionen strukturieren, d.h. welche Kapitelüberschriften hätte es? Zum Vergleich: “XPath and XQuery Functions and Operators 3.1”:

[<https://www.w3.org/TR/xpath-functions-31/>].

- e) Was würden Sie in einer mündlichen Prüfung auf folgende Fragen zu XQuery antworten?
- Was ist die wichtigste Einschränkung von XPath gegenüber XQuery (d.h. was kann man in XQuery machen, was in XPath nicht geht)?
  - Was sind FLWOR-Ausdrücke? Nennen Sie die Schlüsselworte und erklären Sie den Zweck jeder Klausel.
  - Vergleichen Sie FLWOR-Ausdrücke mit SQL. Was sind die Entsprechungen von SELECT und FROM (WHERE ist ja offensichtlich)?
  - Wie sehen direkte Konstruktoren aus? An welchen Stellen darf man auszuwertende Ausdrücke einfügen? Wie werden diese markiert?

- Sei der direkte Konstruktor  $\langle E \rangle \{ Q \} \langle / E \rangle$  gegeben. Die Auswertung der Anfrage  $Q$  liefere eine Sequenz  $S$ . Wie sieht das Ergebnis aus, wenn  $S$  eine Sequenz von atomaren Werten ist? Was passiert, wenn  $S$  am Anfang Attribut-Knoten enthält?
- Was ist “Boundary Whitespace” in einem direkten Konstruktor? Warum ist dieser Begriff wichtig?
- Nennen Sie ein System, mit dem Sie XQuery-Anfragen auswerten können (also eine XML-Datenbank).
- Was ist die aktuelle Version von XQuery?
- Es sei die folgende einfache SQL-Anfrage gegeben:

```
SELECT E.ENAME, E.SAL
FROM   EMP E, DEPT D
WHERE  E.DEPTNO = D.DEPTNO
AND    D.DNAME = 'RESEARCH'
```

Nennen Sie mindestens zwei verschiedene Arten, wie man diese Anfrage nach XQuery übersetzen kann, wenn die Relationen auf eine übliche Art (z.B. mit den Daten in Attributen) in XML repräsentiert sind.

## Für Interessierte

f) Auch die folgenden Webseiten enthalten kürzere oder längere Tutorials:

- Altova: XPath- und XQuery-Tutorial (kurz)  
[<https://www.altova.com/de/mobiletogether/xpath-intro>]
- Dr. Michael Kay: Learn XQuery in 10 Minutes  
[<http://www.stylusstudio.com/xquery-primer.html>]  
[[https://www.stylusstudio.com/docs/v2007/d\\_xquery.html](https://www.stylusstudio.com/docs/v2007/d_xquery.html)]
- Dr. Michael Kay:  
Blooming FLWOR — An Introduction to the XQuery FLWOR Expression  
[<http://www.stylusstudio.com/xquery-flwor.html>]
- Weitere XQuery Tutorials von Dr. Michael Kay  
[<http://www.stylusstudio.com/tutorials/xquery-tutorials.html>]
- Stylus Studio Documentation, Chapter 9: Writing XPath Expressions  
[[https://www.stylusstudio.com/docs/v2007/d\\_xpath.html](https://www.stylusstudio.com/docs/v2007/d_xpath.html)]
- w3schools: XQuery Tutorial  
[[https://www.w3schools.com/xml/xquery\\_intro.asp](https://www.w3schools.com/xml/xquery_intro.asp)]
- XQuery Wikibook  
[<https://en.wikibooks.org/wiki/XQuery>]

g) Hier noch einige Links zu Vorlesungen, die XQuery behandeln:

- Silke Eckstein, Andreas Kupfer (TU Braunschweig): „XML Databases“  
[<http://www.ifis.cs.tu-bs.de/teaching/ws-0910/xmldb>]
- Ehrhard Rahm (Leipzig): DBS 2, Kap. 5: XML-Datenbanken: Datendefinition  
[<https://dbs.uni-leipzig.de/file/dbs2-ss16-kap5.pdf>]
- Ehrhard Rahm (Leipzig): DBS 2, Kap. 6: XML-Datenbanken: Anfragesprachen  
[<https://dbs.uni-leipzig.de/file/dbs2-ss16-kap6.pdf>]
- Ehrhard Rahm (Leipzig): DBS 2, Kap. 7: XML-Datenbanksyst. und SQL/XML  
[<https://dbs.uni-leipzig.de/file/dbs2-ss14-kap7-XML-ORDBS.pdf>]
- Dietmar Seipel (Würzburg): Datenbanken, Kap. 7: XML-Data Modeling  
[[https://www1.pub.informatik.uni-wuerzburg.de/databases/courses/db\\_ws1213/Kapitel\\_7\\_deutsch.pdf](https://www1.pub.informatik.uni-wuerzburg.de/databases/courses/db_ws1213/Kapitel_7_deutsch.pdf)]

h) Schauen Sie sich auch diese Dokumente vom W3C an:

- Don Chamberlin u.a.: XML Query (XQuery) Requirements  
[<https://www.w3.org/TR/xmlquery-req/>]
- Jonathan Robie u.a.: XQuery 3.1: An XML Query Language (W3C Rec., 2017)  
[<https://www.w3.org/TR/xquery-31/>]