

XML und Datenbanken

— 12. Übungsblatt: XSLT —

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, 30.01.2023, 16⁰⁰, über die Übungsplattform in StudIP ab. Schreiben Sie die Lösungen in eine `.xml`- oder `.xslt`-Datei.

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 b) 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 (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.

- a) Schreiben Sie bitte ein XSLT-Stylesheet zur Umwandlung der XML-Daten der CD-Datenbank nach HTML für die folgende Aufgabe.

Das Stylesheet soll eine “unordered list” der CDs erstellen, wobei jeweils

- der Name der CD gedruckt wird,
- anschließend in eckigen Klammern die Anzahl der Aufnahmen auf der CD,
- und dann eine geschachtelte Liste mit den Titeln der Stücke, von denen es Aufnahmen auf der CD gibt.

Die Ausgabe soll also ungefähr so aussehen (die Reihenfolge ist beliebig):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>CD Verzeichnis</title>
  </head>
  <body>
    <h1>CD Liste</h1>
    <ul>
      <li>Händel: Feuerwerksmusik [4]
        <ul>
          <li>Concerto grosso op.6 Nr.7</li>
          <li>Feuerwerks-Musik (Concerto grosso)</li>
          <li>Concerto grosso op.6 Nr.8</li>
          <li>Sinfonia e-moll (aus 'Der Messias')</li>
        </ul>
      </li>
      ...
    </ul>
  </body>
</html>
```

Hinweis: Die Funktion `current()` liefert den aktuellen Knoten, also den, auf den das Template gerade angewendet wird. Der aktuelle Knoten ist zunächst auch der Kontext-Knoten, so dass man in einfachen Mustern diese Funktion nicht braucht. Wenn Sie aber einen etwas komplizierten XPath-Ausdruck schreiben, ändert sich der Kontext-Knoten innerhalb des Ausdrucks (z.B. nach “/” oder innerhalb von “[...]”), während `current()` noch immer den Knoten liefert, auf den das Template angewendet wird. Dies ist z.B. nützlich, wenn Sie von einer CDAufnahme zum zugehörigen `stueck-`Element navigieren wollen.

Wiederholungsaufgaben

- b) Was würden Sie in einer mündlichen Prüfung auf folgende Fragen zu XSLT antworten?
- Eine Anwendung von XSLT ist es, XML Daten-Dateien nach HTML zu übersetzen, um die Daten in einem Web-Browser anzuzeigen. Diese Transformation kann im Browser (Client) oder auf dem Server ausgeführt werden. Was sind die Vor- und Nachteile der beiden Möglichkeiten?
 - Was kann man mit XSLT noch machen, außer XML nach HTML zu übersetzen?
 - Beschreiben Sie den Aufbau eines Templates. Was ist das wichtigste Attribut von `xsl:template`? Welche `xsl`-Elemente werden besonders häufig im Innern von `xsl:template` verwendet?
 - Wenn das Attribut `match` eines Templates den Wert A/B hat, passt es dann auf ein A-Element mit B-Kind, oder auf ein B-Element mit A-Elternteil?
 - Falls mehrere Templates auf einen Knoten passen, welches wird angewendet? Wie kann man ggf. die Auswahl steuern?
 - Was ist der Zweck der eingebauten Templates von XSLT?
 - Was bewirkt `<xsl:apply-templates select="A"/>` im Innern eines Templates, das auf einen Knoten n angewendet wird? D.h. wie wird das entsprechende Stück im Ergebnisbaum bestimmt?
 - Was bewirkt `<xsl:value-of select="A"/>` im Innern eines Templates, das auf einen Knoten n angewendet wird?
 - Wenn ein Template auf den Knoten n angewendet wird, ist dies ja zunächst der Kontext-Knoten. Wie können Sie in Pfad-Ausdrücken im Innern des Templates darauf zugreifen, wenn der Kontext-Knoten verstellt wurde (siehe Aufgabe)?
 - Wenn Sie HTML erzeugen wollen, können Sie im XSLT-Stylesheet dann einfach `
` schreiben, oder müssen Sie `
` schreiben?
 - Wie verweist eine XML-Datei auf ein XSLT-Stylesheet?
 - Was ist der Zweck von `xsl:output`?
 - Ist es möglich, dass ein Stylesheet nicht terminiert? Geben Sie ggf. ein Beispiel.
 - Nennen Sie einige Unterschiede bzw. Einschränkungen von XPath 1.0 gegenüber XPath 2.0.
 - Vergleichen Sie XSLT mit XQuery. Wie werden z.B. neue Knoten konstruiert? Wie kann man sortieren?
 - Wie kann man Fallunterscheidungen in XSLT formulieren? Z.B. wollen Sie bei CDs mit nur einer Aufnahme "1 Aufnahme" (Singular) ausgeben, bei CDs mit mehreren Aufnahmen dagegen z.B. "3 Aufnahmen" (Plural).

Für Interessierte

c) Hier einige Links zu den für XSLT relevanten Standards. Für die praktische Anwendung im Browser sind die „1.0“-Versionen der Standards noch immer besonders relevant.

- XSL Transformations (XSLT) Version 1.0
(W3C Recommendation 16 November 1999)
[<https://www.w3.org/TR/xslt-10/>]
- XSL Transformations (XSLT) Version 2.0 (Second Edition)
(W3C Recommendation 30 March 2021) (Amended by W3C)
[<https://www.w3.org/TR/xslt20/>]
- XSL Transformations (XSLT) Version 3.0
(W3C Recommendation 8 June 2017)
[<https://www.w3.org/TR/xslt-30/>]

Natürlich hängt XSLT direkt mit XPath zusammen:

- XML Path Language (XPath) Version 1.0
(W3C Recommendation 16 November 1999)
[<https://www.w3.org/TR/xpath-10/>]
- XML Path Language (XPath) 2.0 (Second Edition)
(W3C Recommendation 14 December 2010)
[<https://www.w3.org/TR/xpath20/>]
- XML Path Language (XPath) 3.0
(W3C Recommendation 08 April 2014)
[<https://www.w3.org/TR/xpath-30/>]
- XML Path Language (XPath) 3.1
(W3C Recommendation 21 March 2017)
[<https://www.w3.org/TR/xpath-31/>]

Es gibt auch eine Homepage der XSLT-Entwicklung mit Links auf alle relevanten Standards und weiteren Informationen:

- The Extensible Stylesheet Language Family (XSL)
[<https://www.w3.org/Style/XSL/>]

Hier noch die Wikipedia-Seiten mit Hinweisen auf Implementierungen:

- Wikipedia (englisch): XSLT
[<https://en.wikipedia.org/wiki/XSLT>]
- Wikipedia (deutsch): XSL Transformation
[https://de.wikipedia.org/wiki/XSL_Transformation]

d) Im Folgenden einige XSLT Tutorials. Wenn Sie weitere XSLT Lehrmaterialien empfehlen möchten, schicken Sie mir bitte eine EMail. Wenn Sie diese Tutorials als nicht hilfreich empfinden, würde mich das auch interessieren.

- Tutorial von W3 Schools:
[https://www.w3schools.com/xml/xsl_intro.asp]
- Einführung in einem Artikel der Zeitschrift iX:
[<https://www.heise.de/ix/artikel/Mutabor-505896.html>]

e) Leider sind die Folien noch auf dem Stand von XQuery 1.0. In der letzten Übung hatten einige Studierende das „group by“ Konstrukt verwendet, das es in XQuery erst ab der Version 3.0 gibt:

[<https://www.w3.org/TR/xquery-30/#id-group-by>]

Oben auf dieser Webseite (im „Abstract“) sind die neuen Features von XQuery 3.0 aufgelistet.

Übrigens gibt es keine Version 2 von XQuery. XQuery 3.0 (April 2014) ist die nächste offizielle Version nach XQuery 1.0 (Januar 2007). Die aktuelle Version ist XQuery 3.1 (März 2017). In XQuery 3.1 wurde insbesondere die Unterstützung für JSON eingeführt, die JavaScript Object Notation. Die wesentlichen Standards in der Version 3.1 (alle W3C Recommendations vom 21. März 2017) sind:

- XQuery 3.1: An XML Query Language
[<https://www.w3.org/TR/xquery-31/>]
- XQuery and XPath Data Model 3.1
[<https://www.w3.org/TR/xpath-datamodel-31/>]
- XPath and XQuery Functions and Operators 3.1
[<https://www.w3.org/TR/xpath-functions-31/>]
- XML Path Language (XPath) 3.1
[<https://www.w3.org/TR/xpath-31/>]
- XSLT and XQuery Serialization 3.1
[<https://www.w3.org/TR/xslt-xquery-serialization-31/>]

Daneben gibt es noch die „XQuery 3.1 Requirements and Use Cases“ mit vielen Beispielen:

[<https://www.w3.org/TR/xquery-31-requirements/>]

Vielleicht wollen Sie auch die recht kurzen Wikipedia-Artikel zu XQuery lesen:

- [<https://en.wikipedia.org/wiki/XQuery>] (englisch)
- [<https://de.wikipedia.org/wiki/XQuery>] (deutsch)

- f) In der nächsten Vorlesung werden wir uns noch kurz mit JSON beschäftigen. Sie werden JSON in der Praxis immer wieder antreffen. Möglicherweise läuft es XML den Rang ab, weil es so einfach ist und auch zum Austausch von baumartig strukturierten Daten geeignet ist. Besonders für die Client-Server Kommunikation wird JSON gern verwendet, aber es ist auch ein Dateiformat für strukturierte Daten. Der Standard hat nur ca. 6 Seiten:

[https://www.ecma-international.org/wp-content/uploads/ECMA-404_2nd_edition_december_2017.pdf]

Vermutlich die gleichen Syntaxdiagramme finden Sie auch auf folgender Seite:

[<https://www.json.org/json-en.html>]

Davon gibt es auch eine deutsche Version:

[<https://www.json.org/json-de.html>]

Ein kurzes Tutorial finden Sie z.B. auf SelfHTML.org:

[<https://wiki.selfhtml.org/wiki/JSON>]

Für diese Vorlesung brauchen Sie sich aber nicht mit den Einbindungen in Programmiersprachen zu beschäftigen (also z.B. JavaScript selbst). Ein etwas längeres Tutorial gibt es bei w3schools.com, aber auch hier ist der Übergang zur Programmierung in JavaScript (die hier nicht benötigt wird) fließend:

[https://www.w3schools.com/js/js_json_intro.asp]

Die Wikipedia-Artikel zu JSON und verwandten Technologien sind:

- [<https://en.wikipedia.org/wiki/JSON>] (englisch)
- [https://de.wikipedia.org/wiki/JavaScript_Object_Notation] (deutsch)

Die XPath- und XQuery-Einführungen von Altova behandeln auch die neueren Konstrukte. Sie sollten zunächst mindestens die Teile über Arrays und Maps der XPath-3.0 und 3.1 Einführung lesen:

[<https://www.altova.com/training/xpath3>]

Natürlich brauchen Sie die recht langen Listen von Funktionen nicht im Detail durcharbeiten und schon gar nicht auswendig zu lernen.

Die XQuery-Einführung von Altova behandelt auch die neueren Konstrukte:

[<https://www.altova.com/training/xquery3>]

Die BaseX Dokumentation erklärt hier Maps und Arrays:

[https://docs.basex.org/wiki/XQuery_3.1]

Die Beispiele aus dem XQuery-Buch von Priscilla Walmsley sind online verfügbar, für Maps und Arrays z.B.

[<http://www.datypic.com/books/xquery/chapter24.html>]

Allerdings sind einige Beispiele etwas unvollständig.

Falls Sie weitere nützliche Tutorials finden, teilen Sie mir das bitte mit. Ebenso, wenn Sie meinen, man sollte einen der obigen Links streichen.

g) Auch die XML-Unterstützung in relationalen Datenbanken ist noch für die letzte Vorlesung geplant. Zur Unterstützung von XML in PostgreSQL könnten Sie z.B. die folgenden Seiten lesen:

- PostgreSQL Documentation: XML Type
[<https://www.postgresql.org/docs/current/datatype-xml.html>]
- PostgreSQL Documentation: XML Functions
[<https://www.postgresql.org/docs/current/functions-xml.html>]
- Nikolay Samokhvalov: XML Support in PostgreSQL (2007)
[http://ceur-ws.org/Vol-256/submission_15.pdf]
- Peter Eisentraut: PostgreSQL and XML (2008)
[<https://p2d2.cz/files/xml.pdf>]
- PostgreSQL Wiki: XML Support
[https://wiki.postgresql.org/wiki/XML_Support]
- W3 Resource: PostgreSQL XML Functions
[<https://www.w3resource.com/PostgreSQL/postgresql-xml-functions.php>]

h) Hier noch Links zur XML Unterstützung in Oracle und DB2:

- Oracle Application Developer's Guide - XML
[<https://docs.oracle.com/cd/B14504.01/dl/web/B12099.01/adx05xml.htm>]
- pureXML overview – Db2 as an XML database
[<https://www.ibm.com/docs/en/db2/11.5?topic=purexml-overview>]

Besonders DB2 ist sehr stark in der XML Unterstützung.