Prof. Dr. Stefan Brass
Institut für Informatik
MLU Halle-Wittenberg

14. Januar 2017

XML und Datenbanken
— 10. Übungsblatt: XQuery —

Teil a) muss nicht abgegeben werden. Sie sollten sich allerdings zu Hause mit diesen Fragen beschäftigen, um den Stoff der Vorlesung zu wiederholen.

Teil b) ist eine Hausaufgabe. Bitte schicken Sie eine Lösung per EMail an den Dozenten (mit “`xml16`” in der Betreff-Zeile, bis spätestens 21.01.2016, möglichst bis 18.01.2016). Schicken Sie bitte keine leeren EMails nur mit Anhang, diese landen im Spam-Ordner. Nennen Sie die Datei(en) bitte “Nachname_Vorname”, ggf. mit weiterem Suffix.

a) Wie würden Sie in einer mündlichen Prüfung auf folgende Fragen antworten?

- Geben Sie ein Beispiel für einen “computed constructor” für einen Element-Knoten.
- Was kann man mit einem “computed constructor” machen, was mit einem “direct constructor” nicht möglich ist?
- Warum ist es wichtig, dass Knoten aus dem Eingabedokument in den Konstruktoren kopiert werden? Wie kann man nachweisen/ausprobieren, dass dies wirklich geschieht? Geben Sie ein entsprechendes Beispiel.
- Erzeugt die `return`-Klausel immer neue Knoten? Wann werden ggf. Knoten aus dem Eingabedokument geliefert?
- Beschreiben Sie die Syntax von FLWOR-Ausdrücken möglichst präzise. Kann man auch `let` vor `for` verwenden? Kann man `let` und `for` mehrfach verwenden?
- Was ist eine Positionsvariable in der `for`-Klausel? Wie deklariert man sie? Beginnt die Zählung bei 0 oder bei 1?
- Wie gibt man Typen für Variablen unter `for` oder `let` an?
- Wie ist die Reihenfolge der Variablenbelegungen (“Tuple Stream”) definiert, die die `for` und `let`-Klauseln im “ordering mode” `ordered` erzeugen?
- Welche Auswirkungen kann der “Ordering Mode” `unordered` haben?
- Wenn die Dokumentordnung für eine Anfrage wichtig ist, und Sie möglichst portabel programmieren wollen, was müssen Sie dann schreiben (d.h. wie erzwingen Sie den “Ordering Mode” `ordered`)?
- Kann man in der `where`-Klausel nur boolesche Ausdrücke schreiben (Formeln wie in SQL)? Wie wird ggf. ein Wahrheitswert bestimmt?
- Wie funktioniert die `order by`-Klausel? Was muss man beachten, wenn das Dokument nicht validiert wurde?
- Beschreiben Sie den gesamten Ablauf der Auswertung eines FLWOR-Ausdrucks (d.h. die Semantik solcher Ausdrücke).

Hausaufgabe

Verwenden Sie die gleiche Beispiel-Datei wie im letzten Übungsblatt:

[http://www.informatik.uni-halle.de/~brass/xml16/cd.xml]

Schreiben Sie die folgenden Anfragen in XQuery, und probieren Sie diese auch mit einem System aus! Eventuelle Schwierigkeiten und andere interessante Beobachtungen bringen Sie bitte aktiv in der Übung mit ein.

b) Gesucht sind die Namen aller CDs, die Stücke von mehreren Komponisten enthalten (also mindestens 2). Achten Sie darauf, dass jede CD nur einmal ausgegeben wird. Sortieren Sie die Ausgabe nach den CD-Namen. Die Ausgabe soll als “Unordered List” in HTML geschehen, also folgendermaßen aussehen:

```
<ul>
    <li>Corelli,Albinoni,Scarlatti,Manfredini,...</li>
    <li>Leopold Mozart: Sinfonia D-Dur ...</li>
    <li>Mozart/Beethoven: Klassische Ouvertren</li>
    <li>Oboenkonzerte</li>
    <li>Schlager um 1500</li>
    <li>Tschaikowsky/Mendelssohn: Violinkonzerte</li>
</ul>
```

Es sind also nicht die Attribut-Knoten gewünscht, sondern die CD-Namen selbst.

c) Warum ist die Ausgabe folgender Anfrage leer?

```
//komponist[for $s in stueck return true]
```

Tipp: Wenn Ihnen gar nichts einfällt, schauen Sie die Fehlermeldung an, wenn Sie hinter “`true`” noch “`()`” schreiben.

d) Warum liefert die folgende Anfrage viele Duplikate?

```
for $s in //stueck
return distinct-values($/@tonart)
```

e) Schreiben Sie eine XQuery-Anfrage, die alle verschiedenen Tonarten der Datenbank als “unordered list” in HTML ausgibt (wie oben), jede nur einmal, und zuerst die dur-Tonarten, dann die moll-Tonarten. In jedem der beiden Teile (dur, moll) sollen die Tonarten alphabetisch sortiert sein (das ist musikalisch nicht sinnvoll, aber in den Daten fehlt die Information über die Vorzeichen).

Tipp: `contains(s1, s2)` ist eine XPath-Funktion, die prüft, ob `s2` in `s1` als Teilzeichenkette vorkommt.