

XML-Datenbanken Wintersemester 2008/2009 4. Aufgabenblatt

Hinweis: Zur Diskussion der Übungsaufgaben in der Übung sind vorweg formlose Präsentationsfolien zu erstellen. Diese müssen im PDF-Format per Email an kupfer@ifis.cs.tu-bs.de mit dem Betreff "XMLDatenbanken" sowie der Gruppennummer bis spätestens Sonntag vor der Übung abgegeben werden. Achten Sie bitte stets darauf Ihren Namen auf dem Titelblatt zu vermerken.

Aufgabe 1: Vergleichen Sie die in der Vorlesung vorgestellten Verfahren zur Generierung von XML Ergebnissen aus Daten einer relationalen Datenbank miteinander.

Für welches der Verfahren würden Sie sich entscheiden, wenn Sie in einer Übung relationale Daten in eine festgelegte XML-Form bringen sollen. Begründen Sie Ihre Entscheidung.

Aufgabe 2: XSLT kann als Sprache zur Transformation von Anfrageergebnissen in standardisierten XML-Formaten eingesetzt werden. Einige Implementierungen für einen XSLT-Prozessor stehen bereits zur Verfügung.

- Installieren Sie sich Xalan aus dem gleichnamigen Apache Projekt (<http://xalan.apache.org>).
- Führen Sie das vorgestellte Beispiel für eine HTML Ausgabe aus. Die Eingabedateien finden Sie zusätzlich unter http://www.ifis.cs.tu-bs.de/sites/default/files/course/beispiel_zip_16914.zip
- Beschreiben Sie alles notwendige um das Stylesheet aufzurufen.
- Neben der erwarteten Ausgabe befinden sich auch unerwartete Inhalte in der generierten Datei. Welche sind dies und woher stammen sie?
- Ändern Sie das Stylesheet um die unerwünschten Inhalte zu vermeiden.
- Erweitern Sie die Eingabedatei um eine beliebige URL für jede Pizzeria und erweitern Sie das Stylesheet, so dass der Name der Pizzeria als Link mit der richtigen URL anklickbar wird.
- Welche Änderungen sind nötig um zu jeder Pizzeria auch die entsprechende Adresse in der HTML-Datei auszugeben?

Aufgabe 3: XPath Quiz – gegeben sei der Kontext:

```
<a>
  <b><x id="1"/></b>
  <d><x id="2"/><x id="3"/></d>
</a>
```

Vergleichen Sie die folgenden Paare von XPath Ausdrücken. Geben Sie das jeweilige Ergebnis an und erläutern Sie den Unterschied zwischen den beiden Anfragen.

- `(descendant-or-self::node()/child::x)[2]` und `descendant-or-self::node()/child::x[2]`
- `descendant::*[2]` und `descendant::x[2]`
- `//x/@id` und `a//@id`
- `//x[1]` und `descendant-or-self::x[1]`