

## Datenbanksysteme II

### SS 2017 – Übungsblatt 6

#### 1. Aufgabe (XML Schema)

- a) Geben Sie zu der XML-Instanz aus dem Übungsblatt 5 Aufgabe 3 ein konformes XML Schema an für die Elemente *biblioentry*, *authorgroup* und *title* an. Die Elemente *author* und *othercredit* können als *person* verallgemeinert werden. Diesbezüglich soll *person* als komplexer Typ definiert werden und *author* und *othercredit* als Elemente dieses Typs. Die Korrektheit einer XML-Instanz bzgl. eines XML-Schemas lässt sich ebenfalls mit einem Validator überprüfen, z.B. <http://www.xmlforasp.net/SchemaValidator.aspx>.

```
<?xml version="1.0" ?>
<bib>
  <biblioentry id="Rade97">
    <authorgroup>
      <author>
        <surname>Rade</surname>
        <firstname>Lennart</firstname>
      </author>
      <author>
        <surname>Westergren</surname>
        <firstname>Bertil</firstname>
      </author>
      <othercredit>
        <surname>Vachenaer</surname>
        <firstname>Peter</firstname>
        <address>
          <email>pv@springer.de</email>
        </address>
      </othercredit>
    </authorgroup>
    <title>Springers Mathematische Formeln</title>
    <subtitle>Taschenbuch für Ingenieure, Naturwissenschaftler,
              Informatiker, Wirtschaftswissenschaftler</subtitle>
    <publisher>
      <publishername>Springer-Verlag</publishername>
      <address>
        <city>Berlin</city>
      </address>
    </publisher>
    <pubdate>1997</pubdate>
    <pagenums start="33" end="36" />
  </biblioentry>
</bib>
```

## 2. Aufgabe (XMLTable)

Gegeben sei die Tabelle *GolfClubverzeichnis*(Ort, Golfclub). Dabei sei *Golfclub* vom Typ XML

Ort	Golfclub
Leipzig	<pre> &lt;tns:GolfCountryClub xmlns:tns="http://www.example.org/GolfCountryClub" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.example.org/GolfCountryClub GolfCountryClub.xsd "&gt;   &lt;GolfCourse NumberOfHoles="18"&gt;     &lt;Name&gt;Bird's Nest Course&lt;/Name&gt;     &lt;Yardage&gt;7125&lt;/Yardage&gt;     &lt;Par&gt;72&lt;/Par&gt;     &lt;Handicap&gt;74.6&lt;/Handicap&gt;     &lt;Slope&gt;138&lt;/Slope&gt;   &lt;/GolfCourse&gt;   &lt;GolfCourse NumberOfHoles="18"&gt;     &lt;Name&gt;Eagle's Nest Course&lt;/Name&gt;     &lt;Yardage&gt;7198&lt;/Yardage&gt;     &lt;Par&gt;72&lt;/Par&gt;     &lt;Handicap&gt;73.5&lt;/Handicap&gt;     &lt;Slope&gt;136&lt;/Slope&gt;   &lt;/GolfCourse&gt;   &lt;Member handicap="20"&gt;     &lt;RegularMember&gt;       &lt;FirstName&gt;Keith&lt;/FirstName&gt;       &lt;LastName&gt;Chong&lt;/LastName&gt;       &lt;MembershipExpiry&gt;2010-10-31&lt;/MembershipExpiry&gt;     &lt;/RegularMember&gt;   &lt;/Member&gt;   &lt;Member handicap="0"&gt;     &lt;ExclusiveMember&gt;       &lt;FirstName&gt;James&lt;/FirstName&gt;       &lt;MiddleName&gt;Lion&lt;/MiddleName&gt;       &lt;LastName&gt;Chan&lt;/LastName&gt;       &lt;ValetParkingID&gt;124&lt;/ValetParkingID&gt;       &lt;LockerNumber&gt;10&lt;/LockerNumber&gt;     &lt;/ExclusiveMember&gt;   &lt;/Member&gt;   &lt;Member handicap="25"&gt;     &lt;RestrictedMember&gt;       &lt;FirstName&gt;Allan&lt;/FirstName&gt;       &lt;LastName&gt;Chong&lt;/LastName&gt;       &lt;MembershipExpiry&gt;2015-10-31&lt;/MembershipExpiry&gt;       &lt;LimitedPlay&gt;Weekend Only&lt;/LimitedPlay&gt;     &lt;/RestrictedMember&gt;   &lt;/Member&gt;   &lt;Member handicap="10"&gt;     &lt;RegularMember&gt;       &lt;FirstName&gt;Murray&lt;/FirstName&gt;       &lt;MiddleName&gt;Izzun&lt;/MiddleName&gt;       &lt;LastName&gt;Beaton&lt;/LastName&gt;       &lt;MembershipExpiry&gt;2015-09-30&lt;/MembershipExpiry&gt;     &lt;/RegularMember&gt;   &lt;/Member&gt; &lt;/tns:GolfCountryClub&gt; </pre>

Formulieren Sie eine SQL- Query, die alle Mitglieder durch die Angabe von Vorname, Nachname und Handicap in einer Relation ausgibt, die reguläre Mitglieder sind und ein Handicap kleiner als 15 haben. Dabei sei der Golfclub in „Leipzig“.

Skizzieren Sie die resultierende Relation.

### 3. Aufgabe (XPath/XQuery)

Zum Ausführen von XPath/XQuery-Anfragen steht im Rahmen von **LOTS** (Leipzig Online-Test-System) ein Online-XQuery-Trainer unter <http://lots.uni-leipzig.de/xqtrain> bereit. Um im XQuery-Trainer auf einer bestimmten XML-Instanz/Datei Anfragen stellen zu können, ist der Anfrage `doc("datei.xml")` voranzustellen, z.B.:

```
doc("bib.xml")/bib/biblioentry[@id="Rade97"]
```

Nachfolgende Aufgaben beziehen sich auf das Schema der XML-Instanz von Aufgabe 1:

#### 3.1 Erstellen Sie die folgenden Anfragen in *XPath* unter Verwendung von *bib.xml*

- a) Liste aller *ids* von Publikationen
- b) Liste sämtlicher Emailadressen
- c) Liste der Titel aller Publikationen von Verlegern aus Berlin
- d) Liste der *ids* aller Publikationen, in deren Titel oder Untertitel der Ausdruck "wissen" vorkommt
- e) Liste aller Publikationen, die mehr als 2 Seiten umfassen
- f) Liste der Titel der Publikationen, an denen (mindestens) ein Autor beteiligt war, dessen Nachname mit 'W' beginnt
- g) Anzahl aller Publikationen
- h) Liste aller Publikationen, die mehr als einen Autor haben
- i) Liste aller Publikationen, die zwischen 2001 und 2003 erschienen sind

#### 3.2 Erstellen Sie die folgenden Anfragen in *XQuery* unter Verwendung von *bib.xml*. Beispiel:

```
for $b in doc("bib.xml")/bib/biblioentry
where $b/@id="Rade97"
return $b
```

- a) Geben Sie eine alphabetisch sortierte Liste aller Vornamen von Autoren aus.
- b) Geben Sie für jede Publikation den Buchtitel sowie die Autoren aus. Fügen Sie dabei den Vornamen und Nachnamen (mit einem Leerzeichen) zusammen und geben diesen im Element *aname* nach folgendem Muster aus:

```
<pub>
  <title>...</title>
  <aname>...</aname>
  <aname>...</aname>
</pub>
```

c) Geben Sie eine Liste aller Städte zusammen mit den dort ansässigen Verlegern nach folgendem Muster aus:

```
<L>
  <stadt>...</stadt>
  <verleger>...</verleger>
  <verleger>...</verleger>
</L>
```

d) Geben Sie eine aufsteigend sortierte Liste aller Jahre zusammen mit den Titeln der in diesem Jahr veröffentlichten Bücher nach folgendem Muster aus:

```
<year value="2000">
  <title>...</title>
  <title>...</title>
</year>
```

#### 4. Aufgabe (Vorteile NoSQL- Systeme)

Welche Vorteile versprechen NoSQL-Systeme gegenüber relationalen DBS? Nennen Sie für die nachfolgenden Aussagen die Eigenschaft, weshalb NoSQL Systeme sich für diesen Aspekt eignen.

1. Facebook hat 300PB Daten in seinem Data Warehouse<sup>1</sup>.
2. Dabei kommen 600TB Daten täglich neu hinzu.
3. Viele Datenquellen haben kein festvorgegebenes Schema, somit können Objekte eine beliebige Anzahl von Attributen besitzen und beliebige Beziehungen zu anderen Objekten aufweisen.

---

<sup>1</sup> <https://code.facebook.com/posts/229861827208629/scaling-the-facebook-data-warehouse-to-300-pb/>