

Aufgabe 1 – XML-Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  targetNamespace="http://dbs.uni-leipzig.de/ns/bib"
>
<xs:element name="bib">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="bibloentry"
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="bibloentry">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="authorgroup"/>
      <xs:element name="title" type="xs:string"/>
      <xs:element name="subtitle" type="xs:string"
        minOccurs="0"/>
      <xs:element ref="publisher" minOccurs="0"/>
      <xs:element name="pubdate" type="xs:integer"/>
      <xs:element ref="pagenums" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:attribute name="id" use="required"
    type="bibld"/>
</xs:element>
<xs:simpleType name="bibld">
  <xs:restriction base="xs:string">
    <xs:pattern value="[A-Z][A-Za-z]{1,3}[0-9][0-9]" />
  </xs:restriction>
</xs:simpleType>
```

Aufgabe 1 – XML-Schema

```
<xs:complexType name="person">
  <xs:sequence>
    <xs:element name="surname"
      type="xs:string"/>
    <xs:element name="firstname" type="xs:string"
      maxOccurs="unbounded"/>
    <xs:element ref="address"
      minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

```
<xs:element name="authorgroup">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="author"
        type="person"
        maxOccurs="unbounded"/>
      <xs:element name="othercredit"
        type="person"
        minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="address">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="city"
        type="xs:string"
        minOccurs="0"/>
      <xs:element name="email"
        type="xs:string"
        minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Aufgabe 1 – XML-Schema

```
<xs:element name="publisher">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="publishername" type="xs:string"/>
      <xs:element ref="address" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="pagenums">
  <xs:complexType>
    <xs:attribute name="start" use="required"
      type="xs:integer"/>
    <xs:attribute name="end" use="required"
      type="xs:integer"/>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Aufgabe 2 XMLTable

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 15 haben. Dabei sei der Golfclub in „Leipzig“

```
Select X.FirstName, X.LastName, X.Handicap from GolfClubverzeichnis GC,  
XMLTable('//Member/RegularMember[../@handicap<15]' Passing Golfclub  
COLUMNS  
    "Vorname" VARCHAR(100) PATH 'FirstName',  
    "Nachname" VARCHAR(100) PATH 'LastName',  
    "Handicap" INTEGER PATH '../@handicap'  
) AS X  
WHERE GC.Ort = 'Leipzig'
```

Skizzieren Sie die resultierende Relation.

Vorname	Nachname	Handicap
Murray	Beaton	10

Aufgabe 3.2 – XQuery

- a) Geben Sie eine alphabetisch sortierte Liste aller Vornamen von Autoren aus.

```
for $a in doc("bib.xml")//author/firstname
order by $a
return $a
```

Ergebnis:

```
<firstname> Bertil </firstname>
<firstname> Erhard </firstname>
<firstname> Erhard </firstname>
```

...

Duplikatfrei:

```
for $a in fn:distinct-values (doc("bib.xml").. )
```

Aufgabe 3.2 – XQuery

- 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* aus.

```
for $p in doc("bib.xml")//biblioentry return
<pub>{
  $p/title ,
  for $a in $p//author return
    element {"aname"} { fn:concat($a/firstname, " ", $a/surname) }
}</pub>
```

Ergebnis:

```
<pub>
  <title> Springers Mathematische Formeln </title>
  <aname> Lennart Rade </aname>
  <aname> Bertil Westergren </aname>
</pub>
```

Alternativ: `<aname> {text{$a/firstname}} {text{$a/surname}} </aname>`

Falsch: `<aname> {$a/firstname} {$a/surname} </aname>`
→ `<aname><firstname>Lennart</firstname><lastname>Rade</..></..>`

Aufgabe 3.2 – XQuery

- c) Geben Sie eine Liste aller Städte zusammen mit den dort ansässigen Verlegern aus.

```
for $c in fn:distinct-values(doc("bib.xml")//city)
return
  <L>
    <stadt>{ $c }</stadt>
    { for $p in fn:distinct-values(doc("bib.xml")
      //publisher[.//city = $c]/publishername)
      return
        <verleger>{$p}</verleger>
    }</L>
```

Ergebnis:

```
<L>
  <stadt> Berlin </stadt>
  <verleger> Springer-Verlag </verleger>
</L>
```

Aufgabe 3.2 – XQuery

- d) Geben Sie eine aufsteigend sortierte Liste aller Jahre zusammen mit den Titeln der in diesem Jahr veröffentlichten Bücher aus.

```
for $a in fn:distinct-values(doc("bib.xml")/bib/biblioentry/pubdate)
order by $a
return <year value="{ $a }">{
  for $b in doc("bib.xml")/bib/biblioentry[./pubdate/text()=$a]
  return $b/title
}</year>
```

Ergebnis:

```
<year value="2000">
```

```
  <title> Data Warehouse Scenarios for Model Management. </title>
```

```
  ...
```

```
</year>
```