

DATENBANKSYSTEME II

Sommersemester 2020

Übungsblatt 6

Aufgabe 1: XML, DTD

Gegeben sei folgende XML Instanz:

```
<?xml version="1.0"?>
<patientlist>
  <patient id="P_1" sex="2">
    <firstname>Tanmay</firstname>
    <lastname>Patil</lastname>
    <address>
      <city>Leipzig</city>
      <street>Ostwache</street>
      <zip>04318</zip>
    </address>
    <history>
      <diagnosis date="2020-03-04">Vogelgrippe</diagnosis>
      <diagnosis date="2020-03-20">Covid-19</diagnosis>
    </history>
  </patient>
  <patient id="P_2" sex="1">
    <firstname>Arne</firstname>
    <lastname>Johnson</lastname>
    <address>
      <city>Leipzig</city>
      <street>Jahrtausendfeld</street>
      <zip>04177</zip>
    </address>
    <history>
      <diagnosis date="2020-05-01">Kopfschmerzen</diagnosis>
      <diagnosis date="2020-03-20">Magendarmgrippe</diagnosis>
    </history>
  </patient>
</patientlist>
```

Erstellen Sie eine passende DTD. Dabei sollen folgende Bedingungen gelten:

- Eine Patientenliste umfasst beliebig viele Patienten.
- Jeder Patient hat eine Adresse bestehend aus *city*, *street* und *zip*
- Ein Patient hat mindestens eine Diagnose in seiner Historie.
- *id* ist vom Datentyp ID.
- *date* ist ein Pflichtattribut für eine Diagnose.
- *sex* repräsentiert das biologische Geschlecht und ist ein Pflichtattribut. Es kann folgende Werte annehmen: 0=unbekannt, 1= männlich, 2=weiblich und 9=nicht zutreffend.

Aufgabe 2: XML,XML-Schema

Gegeben sei folgendes XML-Schema sowie darauffolgende XML-Instanz. Überprüfen Sie die XML-Instanz, ob diese zu dem XML-Schema valide ist und verbessern Sie diese gegebenenfalls.

```
<?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"
  xmlns:bib="http://dbs.uni-leipzig.de/ns/bib">
  <xs:element name="bib">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="bib:biblioentry" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="biblioentry">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="bib:authorgroup" />
        <xs:element name="title" type="xs:string" />
        <xs:element name="subtitle" type="xs:string"
          minOccurs="0" />
        <xs:element ref="bib:publisher" minOccurs="0" />
        <xs:element name="pubdate" type="xs:date" />
        <xs:element ref="bib:pagenums" minOccurs="0" />
      </xs:sequence>
      <xs:attribute name="id" use="required" type="bib:bibId" />
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="bibId">
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Z][A-Za-z]{1,3}[0-9][0-9]" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="authorgroup">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="author" type="bib:person"
          maxOccurs="unbounded" />
        <xs:element name="othercredit" type="bib:person"
          minOccurs="0"
          maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <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="bib:address" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="publisher">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="publishername" type="xs:string" />
        <xs:element ref="bib: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>
```

```

        </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>
</xs:schema>

```

XML-Instanz

```

<?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>
        <firstname>Peter</firstname>
        <surname>Vachenaer</surname>
        <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"/>
  </biblioentry>
</bib>

```

Aufgabe 3: XML Table

Ein zentrales Krankenhausmanagementsystem verwaltet die Patientenliste aus der Aufgabe 1 innerhalb einer Datenbank mithilfe des Datentyps *XML*.

Ort (lid:INTEGER, Stadt:VARCHAR(100), Bundesland:VARCHAR(100))

Krankenhaus (KNr:INTEGER, lid:INTEGER, Patientenliste:XML)
lid Fremdschlüssel auf Ort

Formulieren Sie eine SQL-Anfrage, die den zeitlichen Verlauf der "Covid-19" Diagnostizierten pro Bundesland ermittelt.

Aufgabe 4: 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("patient_list.xml")//patient[firstname="Arne"]
```

- (a) Erstellen Sie die folgenden Anfragen in XPath unter Verwendung von *patient_list.xml*.
- Liste aller ids von Patienten. (21)
 - Liste sämtlicher Nachnamen. (21)
 - Liste der Nachnamen der Patienten, die in der Postleitzahl 04315. (3) wohnen.
 - Liste der Ids von Patienten, deren Diagnose 'ose' enthält. (4)
 - Liste aller Patienten, die zwischen 2020-04-01 und 2020-05-01 eine 'Covid-19' Erkrankung diagnostiziert bekommen haben.
Hinweis: Verwenden Sie den Datentyp `xs:date` um ein Datum zu erhalten.
 - Anzahl aller weiblichen Patienten.
 - Liste aller Patienten mit mehr als einer Diagnose. (15)

- (b) Erstellen Sie die folgenden Anfragen in XQuery unter Verwendung von *patient_list.xml*.
Beispiel:

```
for $p in (doc("patient_list.xml")//patient)
where $p/firstname="Arne"
return $p
```

- Geben Sie eine alphabetisch sortierte Liste aller Vornamen von Patienten aus.
- Geben Sie für jede Diagnose die entsprechenden Patienten aus, wobei der Vorname und Nachname konkateniert werden sollen. Die Ausgabe soll folgendem Muster entsprechen:

```
<diagnosis>
  <name>...</name>
  <patientname>...</patientname>
  <patientname>...</patientname>
  <patientname>...</patientname>
</diagnosis>
```

- Geben Sie für jede PLZ die Anzahl der Covid-19 Patienten an.

```
<Place>
  <zip>...</zip>
  <number>...</number>
</Place>
```