SSD UE

Übung zu SSD4 Semistrukturierte Datenmodelle und SS 2022, Übung 3

| nistrukturierte Datenmodelle und XML | 33 2022, <b>Ubully</b> 3 |  |  |
|--------------------------------------|--------------------------|--|--|
|                                      | Abgabetermin: 24 03 202  |  |  |

|             | SSD41UE Traxler | Name   | Angelos Angelis | Aufwand in h      | 7 |
|-------------|-----------------|--------|-----------------|-------------------|---|
| $\boxtimes$ | SSD42UE Niklas  |        |                 |                   |   |
|             | SSD43UE Niklas  | Punkte |                 | Kurzzeichen Tutor |   |

### 1. XPath Basics

(4 Punkte – je 0,5 Pkt.)

Übersetzen Sie die nachfolgenden textuellen Angaben in entsprechende **XPath 1.0** –Ausdrücke und führen sie diese auf das Dokument icd10cm.xml aus. Verwenden Sie wo möglich die abgekürzte Schreibweise, falls nicht anders gefordert. Beachten Sie dabei, dass Ihre XPath-Ausdrücke genau den textuellen Angaben entsprechen (*unabhängig* von Ihrem Wissen über die Struktur des XML-Dokuments). Der Kontext für Ihre XPath-Ausdrücke soll die Dokumentwurzel (/) und **nicht** die Elementwurzel (Element ICD10CM.tabular) sein!

- 1. Wählen Sie alle Attribute aus. (Anzahl: 3200)
- 2. Wählen Sie alle Enkelelemente (ohne direkte Kindelemente) von section aus. (*Anzahl: 17413*)
- 3. Wählen Sie die Beschreibung des vorletzten Elements chapter aus.
- 4. Zählen Sie alle Kindknoten (direkte und indirekte) des ersten chapter Elements. (*Anzahl: 5xxx*)
- 5. Zählen Sie alle desc-Elemente aus, die mindestens 20 Zeichen lang sind. (Anzahl: 4xxxx)
- 6. Zählen Sie alle name-Elemente, die mit ,M54' beginnen. (Anzahl: 4x)
- 7. Geben Sie die Beschreibung (desc) der Diagnose mit dem Namen ,M54.50° aus.
- 8. Geben Sie die Beschreibungen (desc) aller Diagnosen aus, die eine Erweiterung (extension) mit den Zeichen "G" und "S" aufweisen. (Anzahl: 56)

# 2. XPath 2.0

**(9 Punkte** – je 1 Punkt**)** 

Anmerkungen: Verwenden Sie für diese Beispiele das XML-Dokument TerminListe.xml. Die Beispiel-Ausgaben dienen zum Vergleich, sind teilweise gekürzt und bestehen aus Ergebnis-Datentyp (links) und Wert (rechts).

2.1 Erstellen Sie eine Liste aller in Terminen verwendeten IDs (d.h. Termin-ID, Ersteller-ID und Ort-ID). Geben Sie jeden Wert nur einmal aus.

```
xs:untypedAtomic 10
xs:untypedAtomic 1
xs:untypedAtomic H1
xs:untypedAtomic 20
[...]
```

2.2 Selektieren Sie alle Serientermine und verketten Sie die Datum- und Uhrzeitattribute des Beginn-Elements mit einem 'T' (zB '2019-03-14T12:00:00'), verwenden Sie dafür for und return.

| xs:string | "2022-03-14T12:00:00" |  |
|-----------|-----------------------|--|

```
xs:string "2022-02-24T15:00:00"
xs:string "2022-03-14T17:00:00"
xs:string "2022-02-28T10:00:00"
xs:string "2022-03-15T12:00:00"
```

2.3 Geben Sie für jeden Termin aus, ob er bereits vorbei ist oder noch ansteht. Vergleichen Sie dazu das Datum des Ende-Elements mit dem aktuellen Datum. Geben Sie die Termin-Beschreibung und einen Info-Text dazu aus ('ist bereits vorbei' bzw. 'steht noch an'). (Termine, die am aktuellen Datum stattfinden, stehen noch an.)

```
xs:string "Vorführung XMLSpy: ist bereits vorbei"
xs:string "Programm Meeting: steht noch an"
xs:string "Burglary Report: ist bereits vorbei"
xs:string "Besprechung Ministerium: steht noch an"
xs:string "Balisage Konferenz: steht noch an"
[...]
```

2.4 Geben Sie die Beschreibung aller Termine aus, die in der XML-Datei nach dem Termin gelistet sind, dessen Beschreibung "Report" enthält.

```
Beschreibung Besprechung Ministerium
Beschreibung Balisage Konferenz
Beschreibung XML Tutorium
Beschreibung Abschlussprüfung
Beschreibung XQuery Q&A
[...]
```

2.5 Prüfen Sie, ob für jeden Termin gilt, dass Beginndatum und -uhrzeit vor dem Endzeitpunkt liegen. Erzeugen Sie dazu aus den Attributen datum und uhrzeit einen dateTime-Datentyp. Verwenden Sie für diese Aufgabe einen quantifizierenden Ausdruck!

```
xs:boolean false
```

2.6 Prüfen Sie, ob es (zumindest) einen verschiebbaren Termin gibt, dessen Beschreibung "XML" oder "Datenbanken" enthält. Verwenden Sie für diese Aufgabe einen quantifizierenden Ausdruck!

```
xs:boolean true
```

2.7 Geben Sie die Beschreibung aller Termine aus, die länger als 90 Minuten dauern. Machen Sie sich dazu mit den Datentypen xs:duration und xs:dayTimeDuration vertraut.

```
Beschreibung Vorführung XMLSpy
Beschreibung Besprechung Ministerium
Beschreibung Balisage Konferenz
Beschreibung XML Tutorium
[...]
```

2.8 Haben alle Serientermine eine Benachrichtigung?

```
xs:boolean false
```

2.9 Geben Sie die Beschreibung von drei kommenden Terminen aus.

```
Beschreibung Programm Meeting
Beschreibung Besprechung Ministerium
Beschreibung Balisage Konferenz
```

3. XQuery (11 Punkte)

Übersetzen Sie die nachfolgenden textuellen Angaben in entsprechende XQuery-Ausdrücke, sodass die bei jeder Aufgabenstellung angegebene Ausgabe erzeugt wird. Testen Sie diese Ausdrücke auf Basis der bereitgestellten XML-Dokumente Entlassungsbrief.xml und icd10cm.xml.

3.1 Selektieren Sie die Diagnose-Elemente, die noch ,offen' sind und sortieren Sie aufsteigend nach dem Datum des Auftretens (von). Umgeben Sie das gesamte Resultat mit einem Element <D>. (1,5 Punkt)

3.2 Selektieren Sie die Diagnose-Elemente, die noch "offen" sind. Fügen Sie anstatt der bisherigen Attribute ein Attribut "aktuell" (mit dem Wert "ja") ein. Das Diagnose-Element soll zwei Kind-Elemente erhalten, eines davon "ICD" mit dem Inhalt des "code"-Attributes, das zweite Kind-Element "Bezeichnung" enthält den Inhalt (Text) der Diagnose. Packen Sie die Ausgabe in ein "D"-Element. Verwenden Sie zur Formatierung der Ausgabe Element- und Attributkonstruktoren. (1,5 Punkte)

3.3 Geben Sie die Arzneimittel in einem Element "Arznei" aus und packen Sie diese in ein Element "Med". Wenn eine Dosierung angegeben ist, transformieren Sie diese in die Attribute "morgens, "mittags", "abends" und "nachts". Sie können davon ausgehen, dass die Dosierung so angegeben wird: M-M-A-N. Ist keine Dosierung angegeben, verwenden Sie das Attribut "hinweis". Als Textinhalt übernehmen Sie den Text des Medikaments. (2 Punkt)

3.4 Erstellen Sie ein Element "Med" und geben Sie darin alle Medikamente in einem Element "Arznei" aus, die eine Diagnose aufweisen. Fügen Sie dem Element "Arznei" als Attribute die Dosierung und den Text der zugehörigen Diagnose hinzu. Die Diagnose können Sie über den "code" ermitteln. Sortieren Sie die Einträge der Arznei alphabetisch. Verwenden Sie dazu den exists-Quantor. (2 Punkte)

</Med>

3.5 Geben Sie die Titel und Vor- und Nachnamen des Patienten in einem Element "Name" aus. Fügen Sie auch das Geburtsdatum als Element "Geburtsdatum" hinzu und berechnen Sie auch das Alter (Attribut "alter"). Umgeben Sie die Elemente mit einem Tag "Person". (2 *Punkte*)

3.6 Verwenden Sie das Dokument icd10cm.xml um auf die offiziellen ICD-Bezeichnungen zuzugreifen. Geben Sie alle Diagnosen des Entlassungsbriefs als leere Elemente aus. Jedes Element soll folgende Attribute aufweisen: code (der angegebene ICD-Code), name (die angegebene Bezeichnung im Entlassungsbrief), desc (die offizielle Bezeichnung, Englisch) und spez (Zählen Sie die Anzahl der Spezialisierungen, dh. Unter-Diagnosen). Das Ergebnis umfassen Sie mit einem Tag "D". (2 Punkte)

# 1.1)

//@\*

```
| type | title | subtitle | subtitle | subtitle | type | instruction | type | t
```

# 1.2)

//section/\*/\*



# 1.3)

//chapter[position() = last() -1]/name



## 1.4)

count(//chapter[1]//\*)

```
# xs:integer 5730
```

# 1.5)

count(//desc[string-length() >= 20])

```
# xs:integer 43223
```

### 1.6)

```
count(//name[contains(text(), 'M54')])
# xs:integer | 40
```

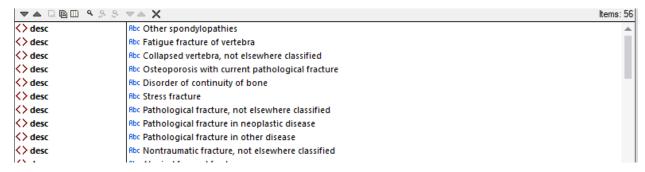
## 1.7)

```
//desc[../name[text() = "M54.50"]]
```

```
<>desc Rbc Low back pain, unspecified
```

### 1.8)

//diag[.//extension[@char = "G"] and .//extension[@char = "S"]]/desc



2)

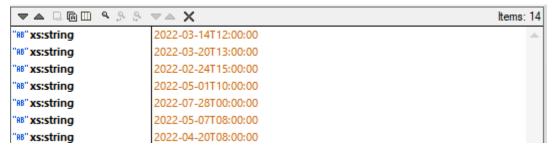
# 2.1)

distinct-values(//\*/@id | //\*/@terminErsteller | //\*/@ort )

```
▼▲□@□٩ṢŞ▼△X
                                                                                                                   Items: 25
"RB" xs:untypedAtomic
"AB" xs:untypedAtomic
                      H1
"AB" xs:untypedAtomic
                      20
"AB" xs:untypedAtomic
"AB" xs:untypedAtomic
"AB" xs:untypedAtomic
                            12
                            120
"#B" xs:untypedAtomic
"AB" xs:untypedAtomic
                            130
                           P1
"#B" xs:untypedAtomic
```

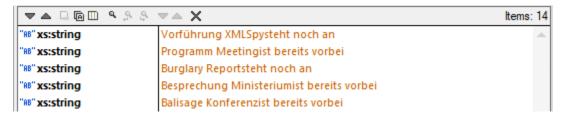
#### 2.2)

```
for $dt in //Termin/Beginn
return
concat($dt/@datum,'T',$dt/@uhrzeit)
```



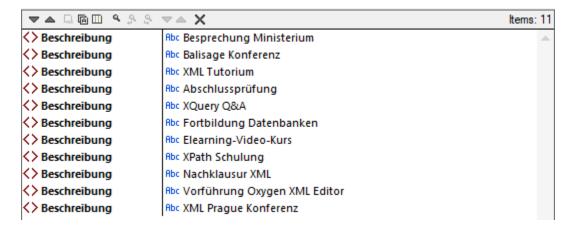
### 2.3)

```
for $ter in //Termin
return
if ( $ter/Ende/@datum < current-date() ) then concat($ter/Beschreibung, 'steht noch
an') else concat($ter/Beschreibung, 'ist bereits vorbei')</pre>
```



## 2.4)

//Termin[contains(./Beschreibung, 'Report')]/following-sibling::\*/Beschreibung



### 2.5)

every \$ter in //Termin satisfies dateTime(\$ter/Beginn/@datum, \$ter/Beginn/@uhrzeit) <
dateTime(\$ter/Ende/@datum, \$ter/Ende/@uhrzeit)</pre>



#### 2.6)

every \$ter in //Termin satisfies contains(\$ter/Beschreibung,'XML') or contains(\$ter/Beschreibung,'Datenbanken')



## 2.7)

for \$ter in //Termin[(dateTime(./Ende/@datum, ./Ende/@uhrzeit) dateTime(./Beginn/@datum, ./Beginn/@uhrzeit)) > xs:dayTimeDuration("PT90M")]
return
\$ter/Beschreibung



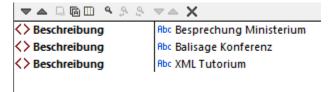
#### 2.8)

every \$ben in //@serienTermin satisfies \$ben[exists(Benachrichtigung)]



### 2.9)

//Termin[dateTime(./Beginn/@datum, ./Beginn/@uhrzeit) > currentdateTime()][position() <= 3]/Beschreibung</pre>



```
3)
3.1)
<D>{
        let $doc := doc("./HUE/Entlassungsbrief.xml")
        for $diag in $doc//Diagnose[./@status = 'offen']
        order by $diag/@von ascending
        return
                $diag
</D>
<D>
   <Diagnose code="M54.9" von="2018-01-31" status="offen">bekannt rezidivierende Rückenschmerzen</Diagnose>
<Diagnose code="G47.63" von="2019-09-19" status="offen">Zähneknirschen</Diagnose>
</D>
3.2)
<D>{
        let $doc := doc("./HUE/Entlassungsbrief.xml")
        for $diag in $doc//Diagnose[./@status = 'offen']
        order by $diag/@von ascending
        return
                <Diagnose>{
                       attribute aktuell{'ja'},
                       <ICD>{data($diag/@code)}</ICD>,
                       <Bezeichnung>{data($diag/text())}</Bezeichnung>
                </Diagnose>
</D>
| <D>
      <Diagnose aktuell="ja">
           <ICD>M54.9</ICD>
           <Bezeichnung>bekannt rezidivierende Rückenschmerzen/Bezeichnung>
      </Diagnose>
      <Diagnose aktuell="ja">
           <ICD>G47.63</ICD>
           <Bezeichnung>Zähneknirschen</Bezeichnung>
      </Diagnose>
```

```
3.3)
<Med>{
       let $doc := doc("./HUE/Entlassungsbrief.xml")
       for $arzn in $doc//Arzneimittel
       return if ( $arzn/@dosierung ne '' ) then
               <Arznei>{
                       attribute morgens{substring(data($arzn/@dosierung),1,1)},
                       attribute mittags{substring(data($arzn/@dosierung),3,1)},
                       attribute abends{substring(data($arzn/@dosierung),5,1)},
                       attribute nachts{substring(data($arzn/@dosierung),7,1)},
                       $arzn/text()
               </Arznei>
               else
               <Arznei>{
                       attribute hinweis{$arzn/@hinweis},
                       $arzn/text()
               }
               </Arznei>
</Med>
<Med>
    <Arznei morgens="1" mittags="0" abends="1" nachts="0">Diazepam Actavis 10 mg Tabletten</Arznei>
<Arznei morgens="1" mittags="0" abends="0" nachts="0">Zithromax 500 mg Filmtabl.</Arznei>
    <Arznei hinweis="3x2 Tropfen">Nasivin 0,05 % - Nasentropfen
    <Arznei morgens="1" mittags="1" abends="1" nachts="0">Seractil</Arznei>
```

```
3.4)
<Med>{
      let $doc := doc("./HUE/Entlassungsbrief.xml")
      for $arzn in $doc//Arzneimittel,
            $diagn in $doc//Diagnose
      return
            <Arznei>{
                  attribute diagnose($diagn/text()),
                  attribute dosierung{$arzn/@dosierung},
                  $arzn/text()
            </Arznei>
</Med>
  3.5)
let $doc := doc("./HUE/Entlassungsbrief.xml")
for $pat in $doc//Patient
return
      <person>{
            <Name>{
            data($pat/Vorname),
            data($pat/Nachname)
            }
            </Name>,
            <Geburtsdatum>{
                  attribute alter{days-from-duration(current-date()-
xs:date($pat/Geburtsdatum)) div 356},
                  $pat/Geburtsdatum/text()
            </Geburtsdatum>
      </person>
    <Name>Konrad Anton Tneitap</Name>
    <Geburtsdatum alter="67.27247191011235955056179775280899">1956-08-24</Geburtsdatum>
</person>
```