Grundlagen von Datenbanken

Aufgabenzettel 7

XML, XSD, XPath

XML-Schemadefinition mit XSD (1)

- Schemadefinition für XML
  - Schema schränkt die für das Dokument erlaubten Strukturen und Datentypen ein
- Dokumentaufbau

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.personen.org"  xmlns ="http://www.personen.org" >
                                        ...
 </xsd:schema>
```

- Namensräume
  - Zur eindeutigen Identifizierung von Elementen

XML-Schemadefinition mit XSD (2)

- Einfache Elementtypen

```
<xsd:element name="Straße" type="xsd:string" minOccurs="1" maxOccurs="1" />
```

- Komplexe Elementtypen

```
<xsd:element name="Anschrift">
    <xsd:complexType>
      <xsd:choice>
        <xsd:element name=„Adresse" type="AdresseTyp" minOccurs="1" maxOccurs="1" />
        <xsd:element name=„Postfach" type="xsd:string" minOccurs="1" maxOccurs="1" />
      </xsd:choice>
    </xsd:complexType>
</xsd:element>
```

- Datentypen können benutzerdefiniert sein

```
<xsd:complexType name="AdresseTyp " mixed="true" >
      <xsd:sequence>
            <xsd:element name="Straße" type="xsd:string" minOccurs="1" maxOccurs="1" />
            <xsd:element name="PLZ" type="PLZTyp" minOccurs="1" maxOccurs="1" />
            <xsd:element name="Stadt" type="xsd:string" minOccurs="1" maxOccurs="1" />
      </xsd:sequence>
</xsd:complexType>
```

XML-Schemadefinition mit XSD (3)

- Identifier
  - ID: dokumentweit und typübergreifend eindeutiger Identifier
  - Key: dokumentweit und typintern eindeutiger Identifier
- Referenzen
  - IDREF: referenziert zur ID eines Elementes
    Problem: nicht getypt
    => bei einer Referenz von einer Firma in dem Attribut ‚arbeitet_fuer' des Elementes ‚Person' könnte die ID einer anderen Person stehen.
  - KeyREF: referenziert zu einem anderem Key

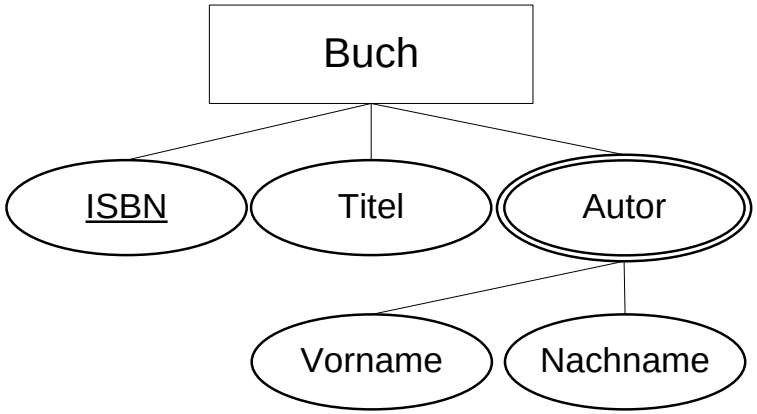
```
<xsd:complexType name="student„
      ….
      <xsd:attribute name=„mnr" type="xs:integer"/>
</xsd:complexType>

<xsd:complexType name=„pruefung„
      …
      <xsd:attribute name=„pruefling" type="xs:integer"/>
</xsd:complexType>
```

```
<xsd:key name="idKeyStudent">
      <xsd:selector xpath=„student"/>
      <xsd:field xpath="@mnr"/>
</xsd:key>
```

```
<xsd:keyref name="idFremdKey"
            refer="idKeyStudent">
      <xsd:selector xpath=„pruefung"/>
      <xsd:field xpath="@pruefling"/>
</xsd:keyref>
```

Abbildung ERD-XSD

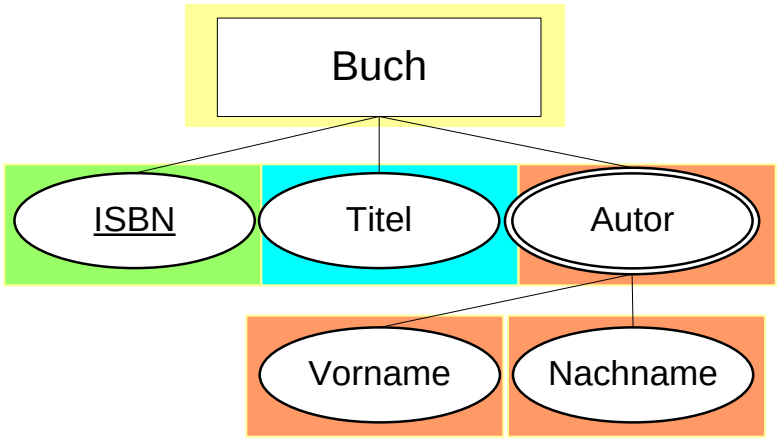
- Entities
  - 1:1-Abbildung auf XML-Elemente
  - <Key> (<KeyREF>) oder <ID> (<IDREF>) in XML-Schema, um (Fremd-)Schlüssel darzustellen




```
<xsd:element name=„Buch„ minOccurs=„0" maxOccurs=„unbounded“>
      <xsd:complexType>
            <xsd:sequence>
                <xsd:element name=„Titel" type="xsd:string„/>
                 <xsd:element name=„Autor„  type="xsd: AutorTyp„
                       minOccurs="1" maxOccurs=„unbounded„/>
            </xsd:sequence>
            <xsd:attribute name=„ISBN" type="xsd:ID" />
      </xsd:complexType>
</xsd:element>
<xsd:complexType name=“AutorTyp” mixed=“true” >
      <xsd:sequence>
            <xsd:element name=“Vorname” type=“xsd:string” />
            <xsd:element name=“Nachname” type=“xsd:string” />
      </xsd:sequence>
</xsd:complexType>
```

Abbildung ERD-XSD

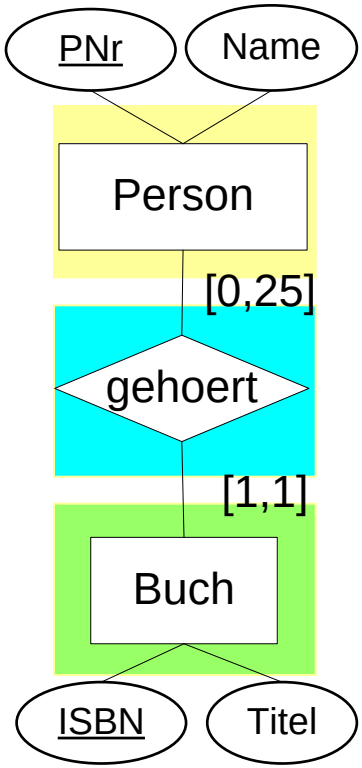
- Entities
  - 1:1-Abbildung auf XML-Elemente
  - <Key> (<KeyREF>) oder <ID> (<IDREF>) in XML-Schema, um (Fremd-)Schlüssel darzustellen




```
<xsd:element name=„Buch„ minOccurs=„0" maxOccurs=„unbounded“>
      <xsd:complexType>
            <xsd:sequence>
                  <xsd:element name=„Titel" type="xsd:string„/>
                  <xsd:element name=„Autor„  type="xsd: AutorTyp„
                        minOccurs="1" maxOccurs=„unbounded„/>
            </xsd:sequence>
            <xsd:attribute name=„ISBN" type="xsd:ID" />
      </xsd:complexType>
</xsd:element>
<xsd:complexType name=“AutorTyp” mixed=“true” >
      <xsd:sequence>
            <xsd:element name=“Vorname” type=“xsd:string” />
            <xsd:element name=“Nachname” type=“xsd:string” />
      </xsd:sequence>
</xsd:complexType>
```

Abbildung ERD-XSD

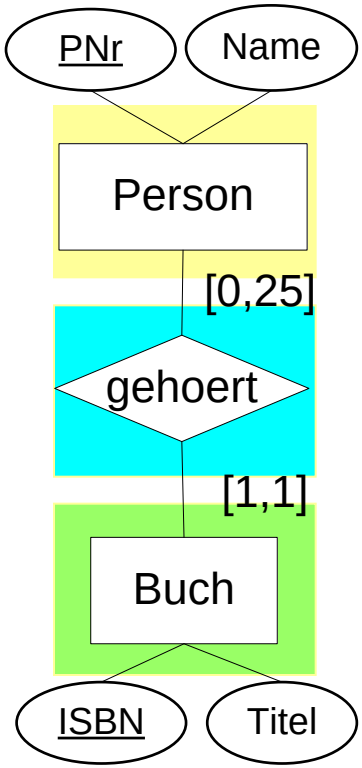
- Relationships (1:1, 1:N)
  - Schachtelung von Elementen
    - Probleme: Existenzabhängigkeit, Eindeutigkeit von Schlüsselattributen in geschachtelten




```
<xsd:element name=„Person„ minOccurs=„0" maxOccurs=„unbounded“ >
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name=„Name" type=" xsd:string „/>
            <xsd:element name=„Buch„ minOccurs=„0" maxOccurs=„25">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name=„Titel" type=" xsd:string „/>
                    </xsd:sequence>
                    <xsd:attribute name=„ISBN" type="xsd:ID" />
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
        <xsd:attribute name=„PNr" type="xsd:ID" />
    </xsd:complexType>
</xsd:element>
```

Abbildung ERD-XSD

- Relationships (1:1, 1:N)
  - Schachtelung von Elementen
    - Probleme: Existenzabhängigkeit, Eindeutigkeit von Schlüsselattributen in geschachtelten

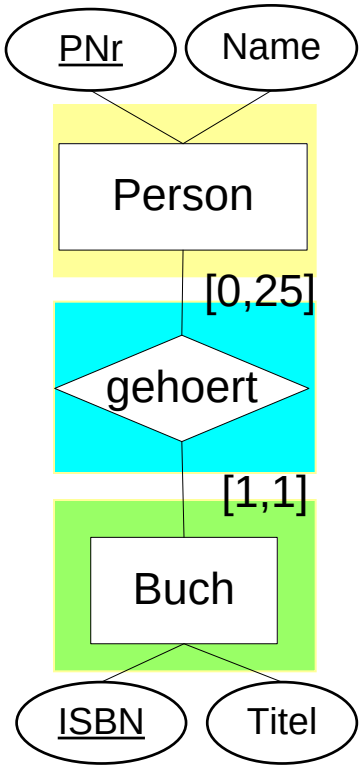



```
<xsd:element name=„Person„ minOccurs=„0" maxOccurs=„unbounded“ >
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name=„Name" type=" xsd:string „/>
            <xsd:element name=„Buch„ minOccurs=„0" maxOccurs=„25">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name=„Titel" type=" xsd:string „/>
                    </xsd:sequence>
                    <xsd:attribute name=„ISBN" type="xsd:ID" />
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
        <xsd:attribute name=„PNr" type="xsd:ID" />
    </xsd:complexType>
</xsd:element>
```

Wird durch die Verschachtelung von Buch in Person modelliert

Abbildung ERD-XSD

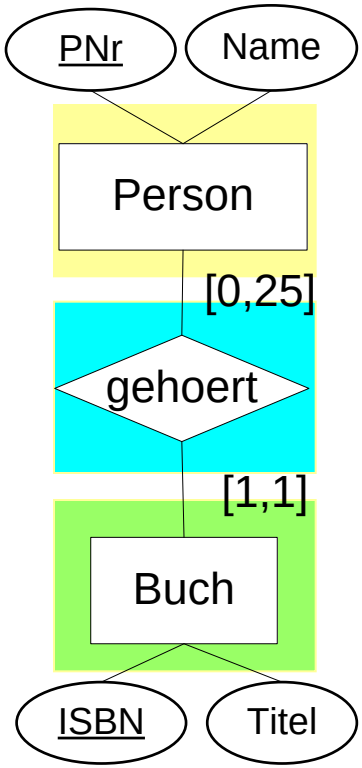
- Relationships (1:1, 1:N)
  - „flache" Repräsentation
    - Nutzung von Schlüsseln, Fremdschlüsseln




```
<xsd:element name=„Person„ >
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name=„Name" type=" xsd:string „/>
         </xsd:sequence>
         <xsd:attribute name=„PNr" type="xsd:ID" />
      </xsd:complexType>
</xsd:element>
<xsd:element name=„Buch„>
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name=„Titel" type=" xsd:string „/>
         </xsd:sequence>
         <xsd:attribute name=„ISBN" type="xsd:ID" />
         <xsd:attribute name=„gehoert" type="xsd:IDREF" />
      </xsd:complexType>
</xsd:element>
```

Abbildung ERD-XSD

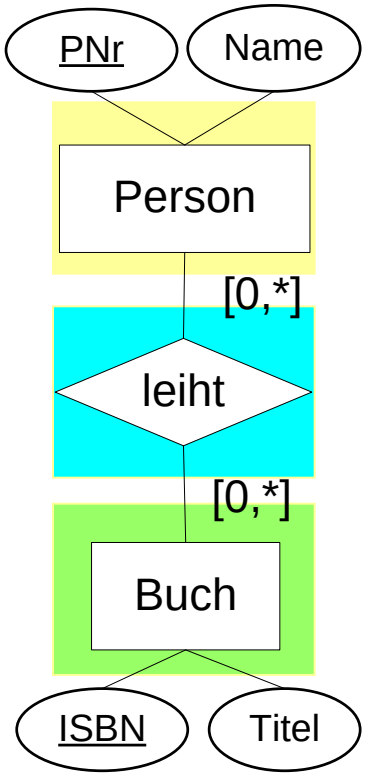
- Relationships (1:1, 1:N)
  - „flache" Repräsentation
    - Nutzung von Schlüsseln, Fremdschlüsseln




```
<xsd:element name=„Person„ >
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name=„Name" type=" xsd:string „/>
      </xsd:sequence>
      <xsd:attribute name=„PNr" type="xsd:ID" />
    </xsd:complexType>
</xsd:element>
<xsd:element name=„Buch„>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name=„Titel" type=" xsd:string „/>
      </xsd:sequence>
      <xsd:attribute name=„ISBN" type="xsd:ID" />
      <xsd:attribute name=„gehoert" type="xsd:IDREF" />
    </xsd:complexType>
</xsd:element>
```

Abbildung ERD-XSD

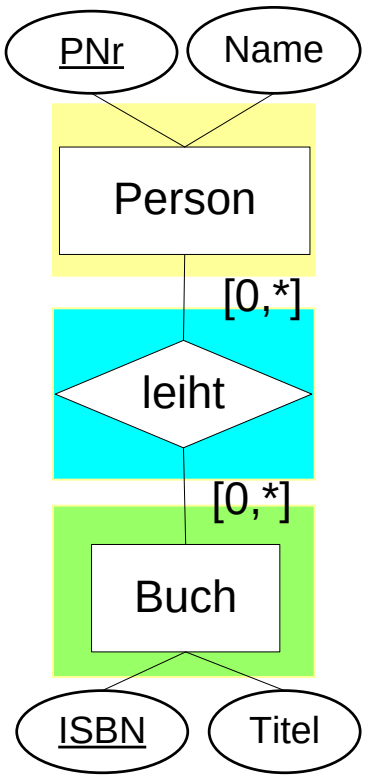
- Relationships (N:M)
  - flache Repräsentation mit Hilfselementen




```
<xsd:element name=„Person„ minOccurs=„0" maxOccurs=„unbounded“ >
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name=„Name" type="xsd:string“/>
        </xsd:sequence>
        <xsd:attribute name=„PNr" type="xsd:ID" />
    </xsd:complexType>
</xsd:element>
<xsd:element name=„Buch„ minOccurs=„0" maxOccurs=„unbounded“>
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name=„Titel" type="xsd:string“/>
        </xsd:sequence>
        <xsd:attribute name=„ISBN" type="xsd:ID" />
    </xsd:complexType>
</xsd:element>
<xsd:element name=„leiht_aus„ minOccurs=„0" maxOccurs=„unbounded“>
    <xsd:complexType>
        <xsd:attribute name=„person" type="xsd:IDREF" />
        <xsd:attribute name=„buch" type="xsd:IDREF" />
    </xsd:complexType>
</xsd:element>
```

Abbildung ERD-XSD

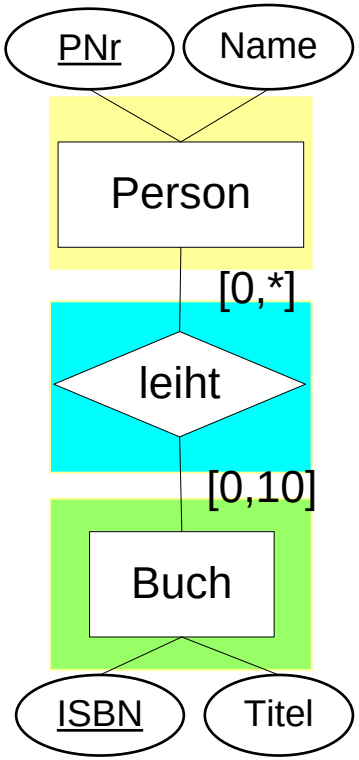
- Relationships (N:M)
  - flache Repräsentation mit Hilfselementen




```
<xsd:element name=„Person„ minOccurs=„0" maxOccurs=„unbounded“ >
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name=„Name" type="xsd:string“/>
        </xsd:sequence>
        <xsd:attribute name=„PNr" type="xsd:ID" />
    </xsd:complexType>
</xsd:element>
<xsd:element name=„Buch„ minOccurs=„0" maxOccurs=„unbounded“>
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name=„Titel" type="xsd:string“/>
        </xsd:sequence>
        <xsd:attribute name=„ISBN" type="xsd:ID" />
    </xsd:complexType>
</xsd:element>
<xsd:element name=„leiht_aus„ minOccurs=„0" maxOccurs=„unbounded“>
    <xsd:complexType>
        <xsd:attribute name=„person" type="xsd:IDREF" />
        <xsd:attribute name=„buch" type="xsd:IDREF" />
    </xsd:complexType>
</xsd:element>
```

Abbildung ERD-XSD

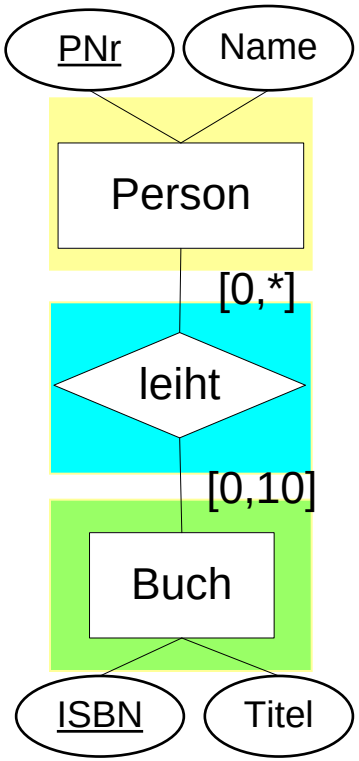
- Relationships (N:M)
  - In einigen Fällen ist aber auch eine Kombination aus ‚flach' und ‚geschachtelt' sinnvoll




```
<xsd:element name="Person" minOccurs="0" maxOccurs="unbounded">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="name" type="xsd:string" />
        </xsd:sequence>
        <xsd:attribute name="PNr" type="xsd:ID" />
    </xsd:complexType>
</xsd:element>
<xsd:element name="Buch" minOccurs="0" maxOccurs="unbounded">
     <xsd:complexType>
          <xsd:sequence>
              <xsd:element name="titel" type="xsd:string" />
              <xsd:element name="leihende_Person" minOccurs="0" maxOccurs="10">
                  <xsd:complexType>
                      <xsd:attribute name="PNr" type="xsd:IDREF" />
                  </xsd:complexType>
              </xsd:element>
          </xsd:sequence>
          <xsd:attribute name="ISBN" type="xsd:ID" />
     </xsd:complexType>
</xsd:element>
```

Abbildung ERD-XSD

- Relationships (N:M)
  - In einigen Fällen ist aber auch eine Kombination aus ‚flach' und ‚geschachtelt' sinnvoll




```
<xsd:element name="Person" minOccurs="0" maxOccurs="unbounded">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="name" type="xsd:string" />
        </xsd:sequence>
        <xsd:attribute name="PNr" type="xsd:ID" />
    </xsd:complexType>
</xsd:element>
<xsd:element name="Buch" minOccurs="0" maxOccurs="unbounded">
     <xsd:complexType>
          <xsd:sequence>
               <xsd:element name="titel" type="xsd:string" />
               <xsd:element name="leihende_Person" minOccurs="0" maxOccurs="10">
                    <xsd:complexType>
                         <xsd:attribute name="PNr" type="xsd:IDREF" />
                    </xsd:complexType>
               </xsd:element>
          </xsd:sequence>
          <xsd:attribute name="ISBN" type="xsd:ID" />
     </xsd:complexType>
</xsd:element>
```

Abbildung ERD-XSD (Zusammenfassung)

- Entities
  - 1:1-Abbildung auf XML-Elemente
  - <Key> (<KeyREF>) oder <ID> (<IDREF>) in XML-Schema, um (Fremd-)Schlüssel darzustellen

- Relationships
  - 1:1, 1:N
    - Schachtelung von Elementen
      - Probleme: Existenzabhängigkeit, Eindeutigkeit von Schlüsselattributen in geschachtelten Elementen, semantische Integritätsbedingungen
    - „flache" Repräsentation
      - Nutzung von Schlüsseln, Fremdschlüsseln
  - N:M
    - flache Repräsentation mit Hilfselementen
    - In einigen Fällen aber auch eine Schachtelung möglich

Anfragen mit XPath

- Pfadausdruck adressiert (selektiert) eine Sequenz von Knoten in einem Dokument
  - besteht aus Schritten, durch "/" voneinander getrennt
    - Bsp.: Namen aller Personen

      /child::Personen/child::Person[Alter>18]/attribute::Name
  - Verkürzte Schreibweisen für häufig genutzte Konstrukte
    ("@"= attribute::, "//"= /descendant-or-self::node()/, Achse fehlt= child::, etc.)
    - Bsp.: Namen aller Personen

      //Person[Alter>18]/@Name
  - wird sukzessive, von links nach rechts ausgewertet
    - Pfadanfang: Dokumentwurzel oder von außen vorgegebener Kontext
    - Jeder Schritt geht von Knotensequenz aus
      - Sucht für jeden Knoten in der Sequenz weitere Knoten auf
      - Prädikate (z.B. [Alter>18]) sorgen dabei für eine Knotenselektion
    - Leere Resultate führen nicht zu Fehler

Anfragen mit XPath

- Nutzung von Funktionen
  - fn:id = löst IDREF-Referenzen auf und gibt referenzierte Elemente zurück
  - fn:count = zählt die selektierten Elemente

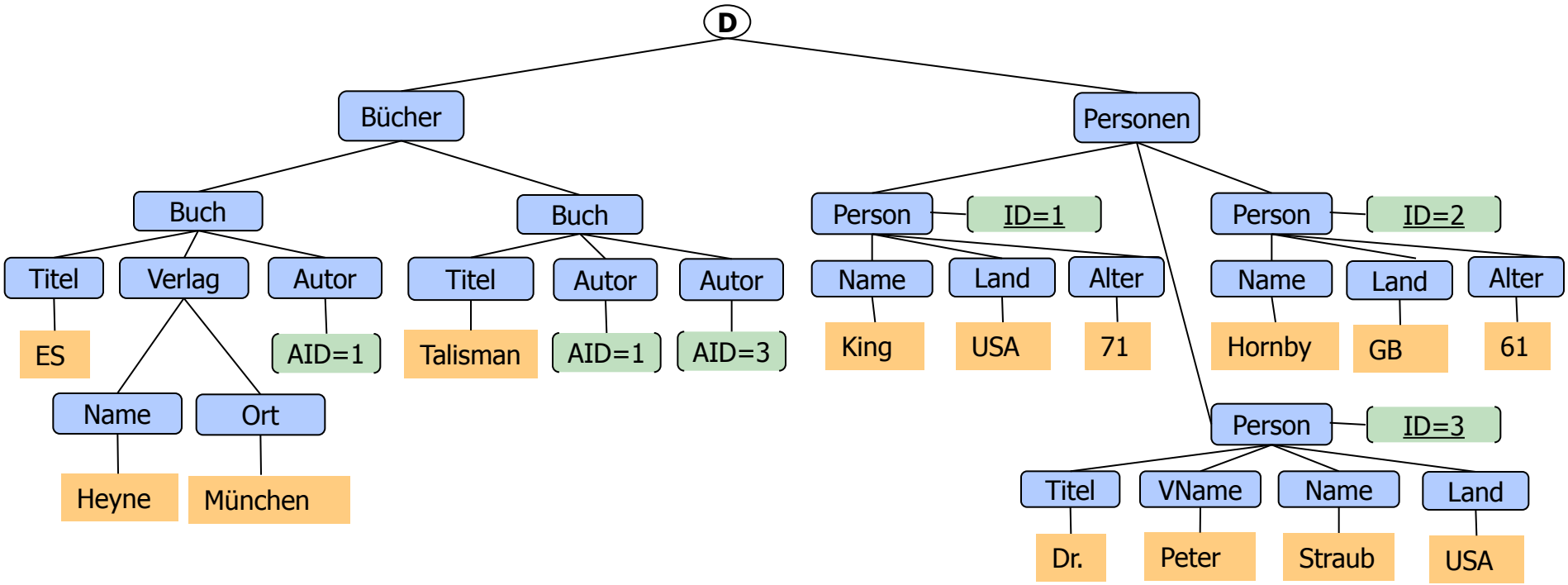
- Beispiele für XPath mit und ohne Funktionen:
  - //person[anschrift/ort/text()=‚Hamburg‘]/Nachname

    „Die Nachnamen aller Personen, die in Hamburg wohnen.“
  - fn:id(//verhandlung[@datum='18.10.2010']/zeuge/@person)/Nachname

    „Die Nachnamen aller Zeugen, die an Verhandlungen am 18.10.2010 teilgenommen haben.“
  - fn:count(//person[@pID = //zeuge/@person and @alter < 18])

    “Die Anzahl der Zeugen, die jünger als 18 sind.”

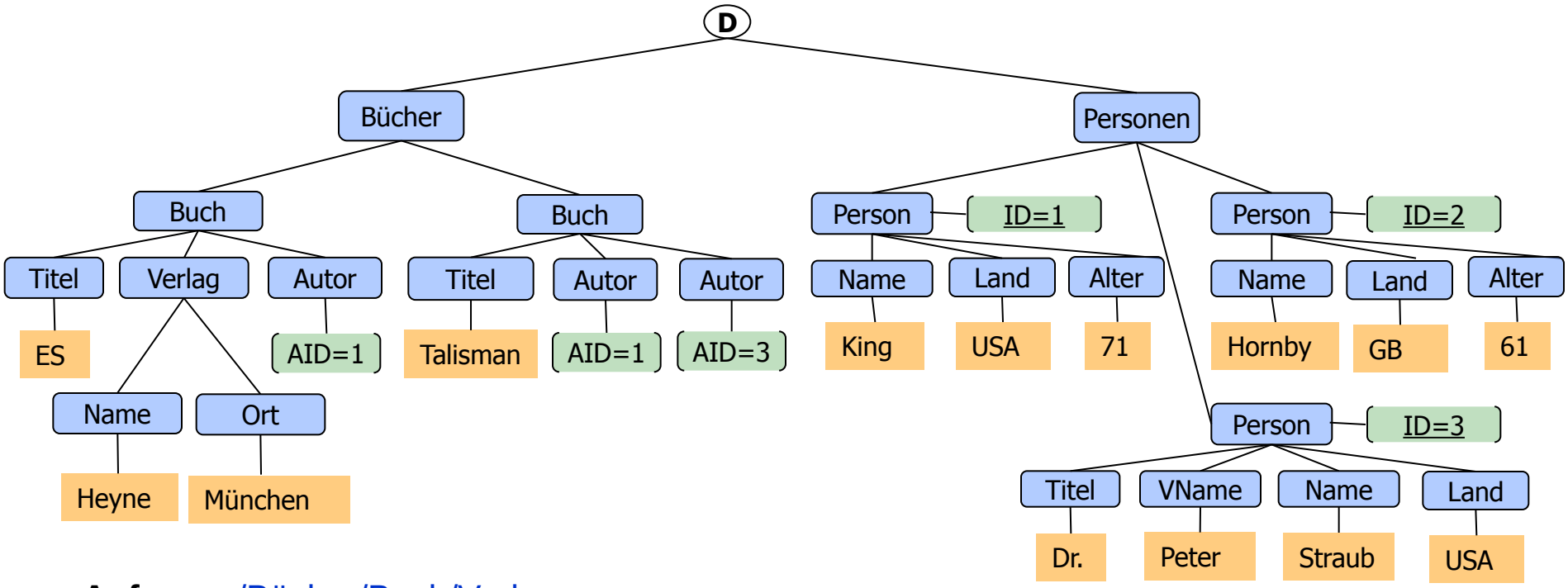
XPath - Beispielaufgaben



Legende:

Element	Attribut
Textdaten	ID Attribut

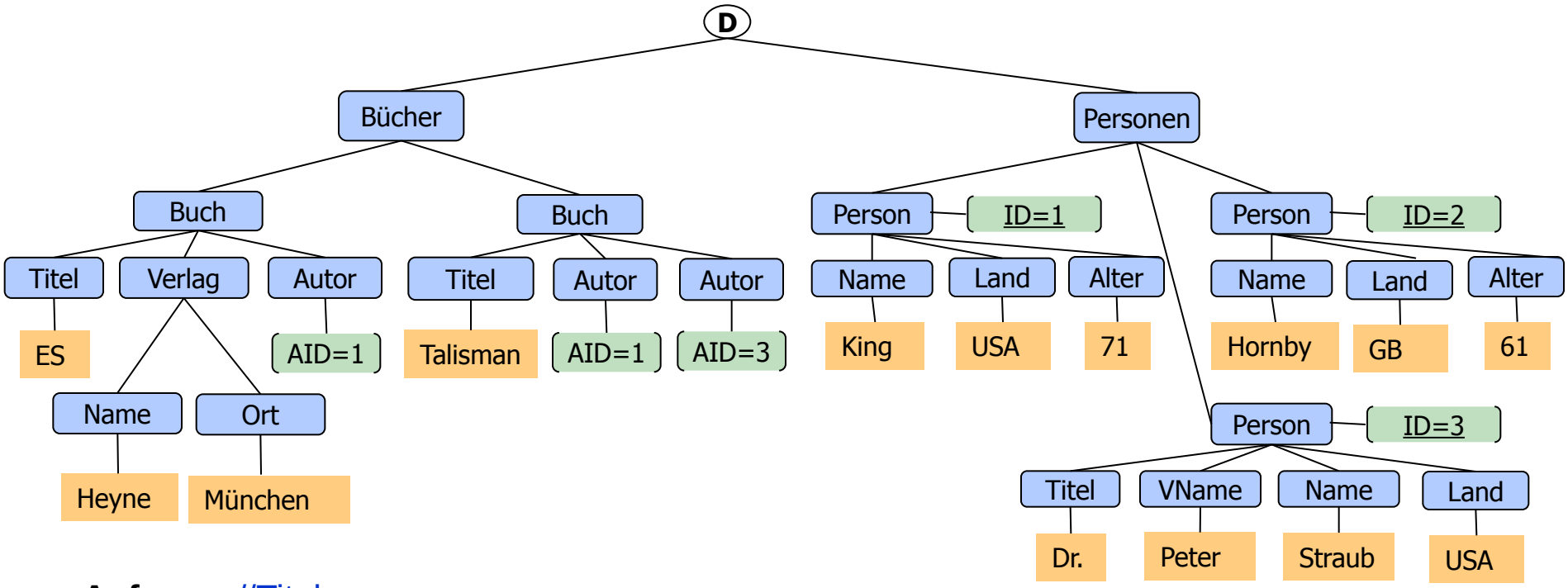
XPath - Beispielaufgaben



Anfrage: /Bücher/Buch/Verlag

Ergebnis:
```
<Verlag>
    <Name>Heyne</Name>
    <Ort>München</Ort>
</Verlag>
```

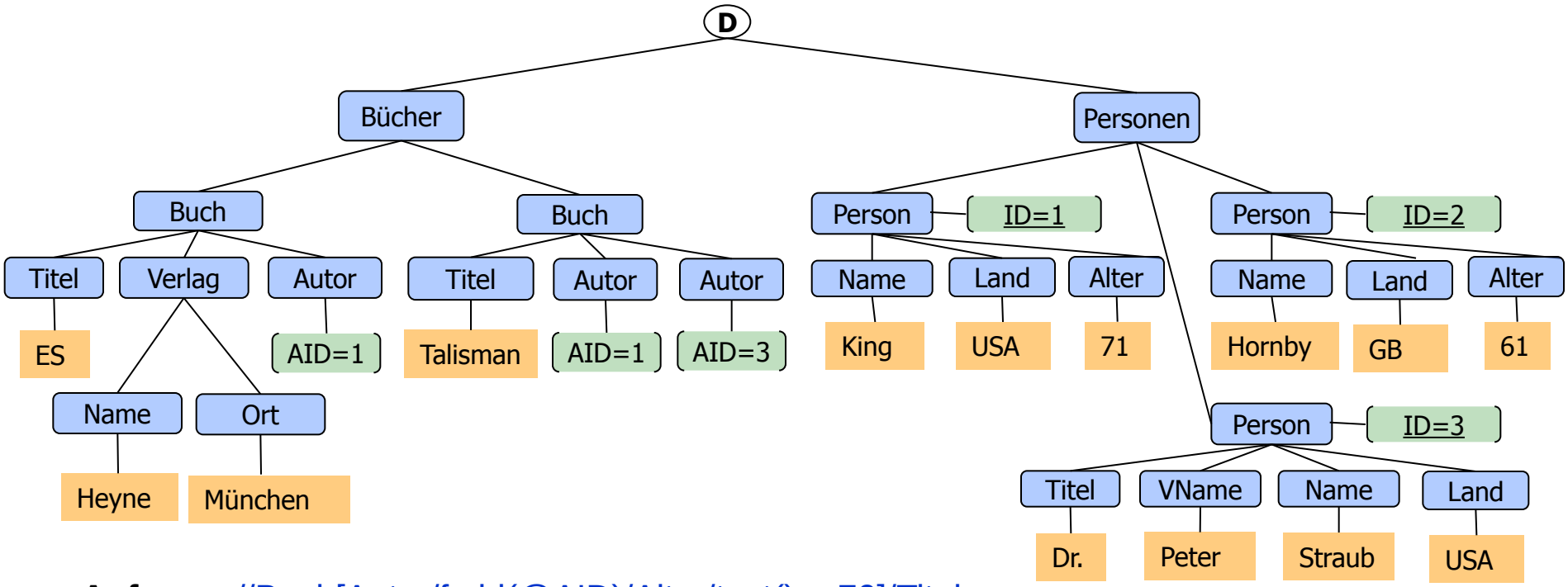
XPath - Beispielaufgaben



Anfrage: //Titel

Ergebnis: <Titel>ES</Titel>
<Titel>Talisman</Titel>
<Titel>Dr.</Titel>

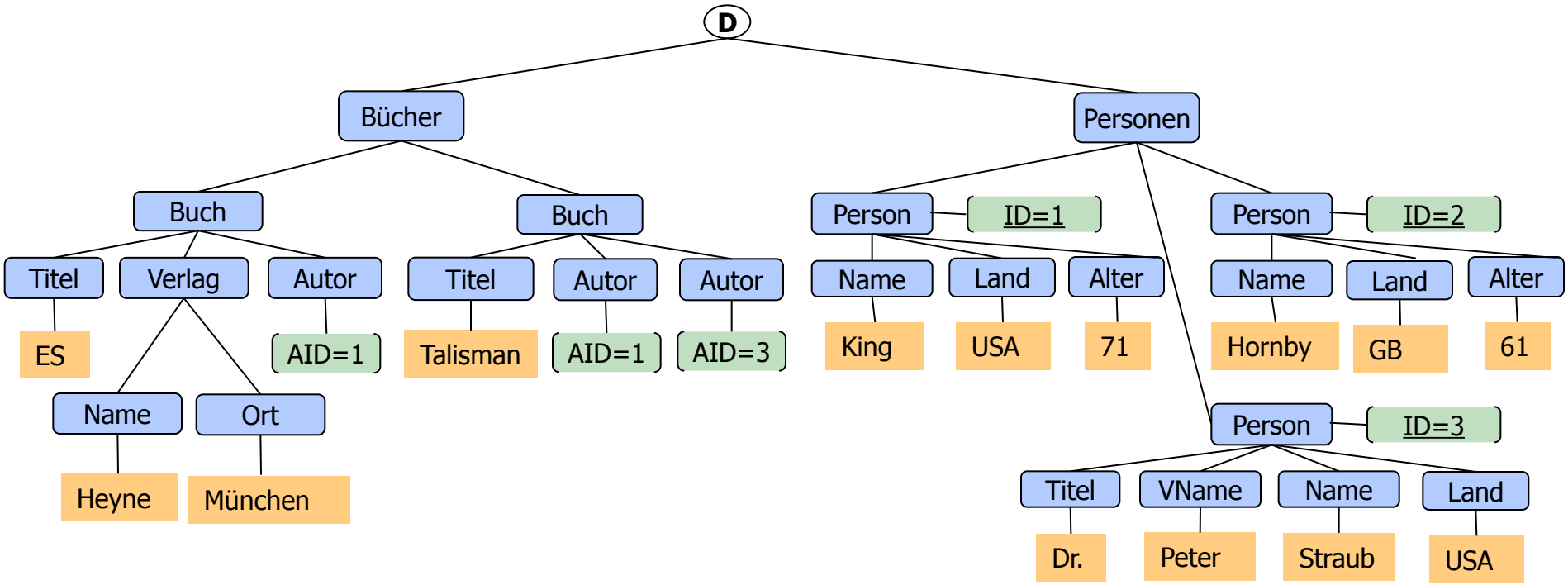
XPath - Beispielaufgaben



Anfrage: //Buch[Autor/fn:id(@AID)/Alter/text() > 70]/Titel

Ergebnis: <Titel>ES</Titel>
<Titel>Talisman</Titel>

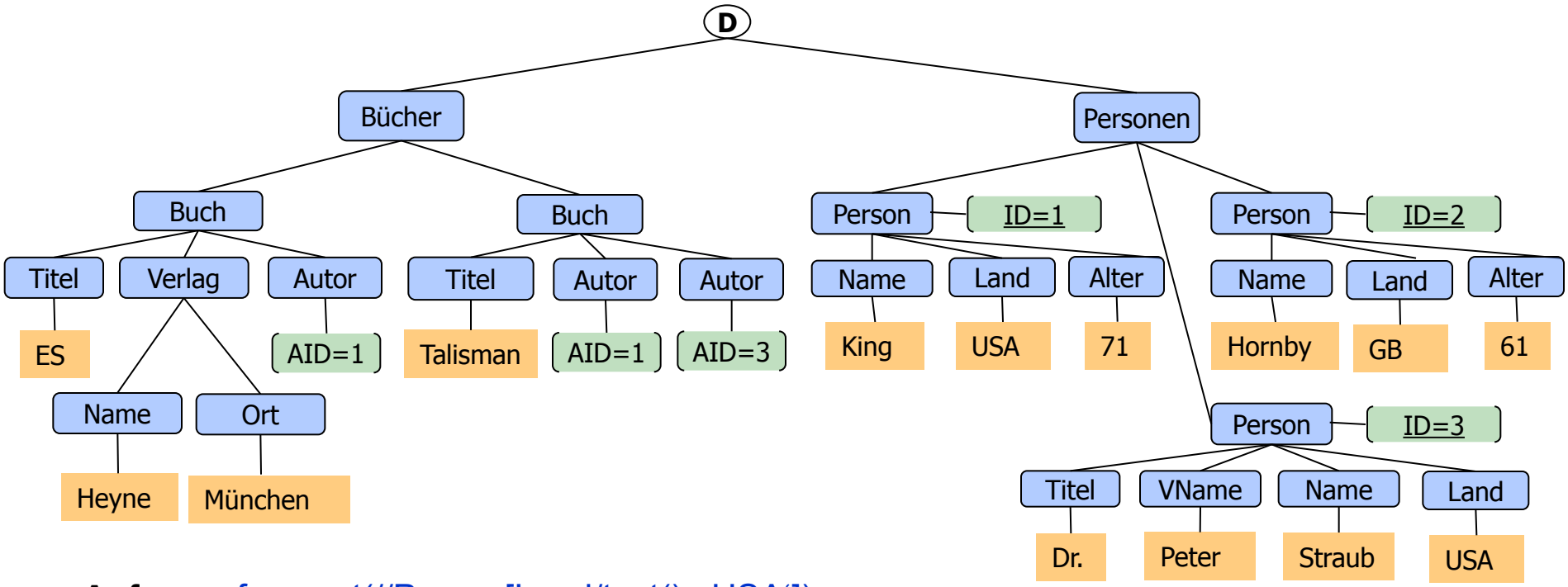
XPath - Beispielaufgaben



Anfrage: //Buch[Verlag/Name/text()=‚Heyne']/Autor/fn:id(@AID)/*

Ergebnis:
```
<Name>King</Name>
<Land>USA</Land>
<Alter>71</Alter>
```

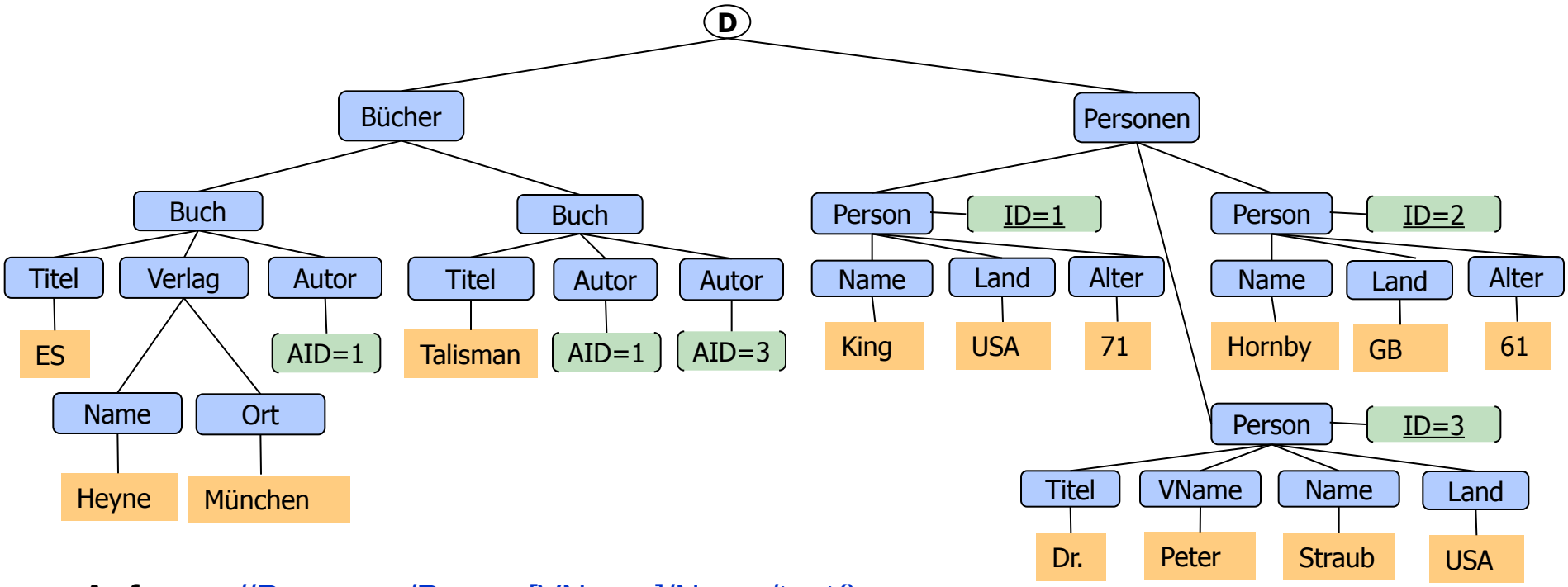
XPath - Beispielaufgaben



Anfrage: fn:count(//Person[Land/text()=‚USA‘])

Ergebnis: 2

XPath - Beispielaufgaben



Anfrage: //Personen/Person[VName]/Name/text()

Ergebnis: Straub