



Einsatz und Realisierung von Datenbanksystemen

ERDB Übungsleitung

Alice Rey, Maximilian Bandle, Michael Jungmair

i3erdb@in.tum.de



Organisatorisches

Disclaimer

Die Folien werden von der Übungsleitung allen Tutoren zur Verfügung gestellt.

Sollte es Unstimmigkeiten zu den Vorlesungsfolien von Prof. Kemper geben, so sind die Folien aus der Vorlesung ausschlaggebend.

Falls Ihr einen Fehler oder eine Unstimmigkeit findet, schreibt an i3erdb@in.tum.de mit Angabe der Foliennummer.

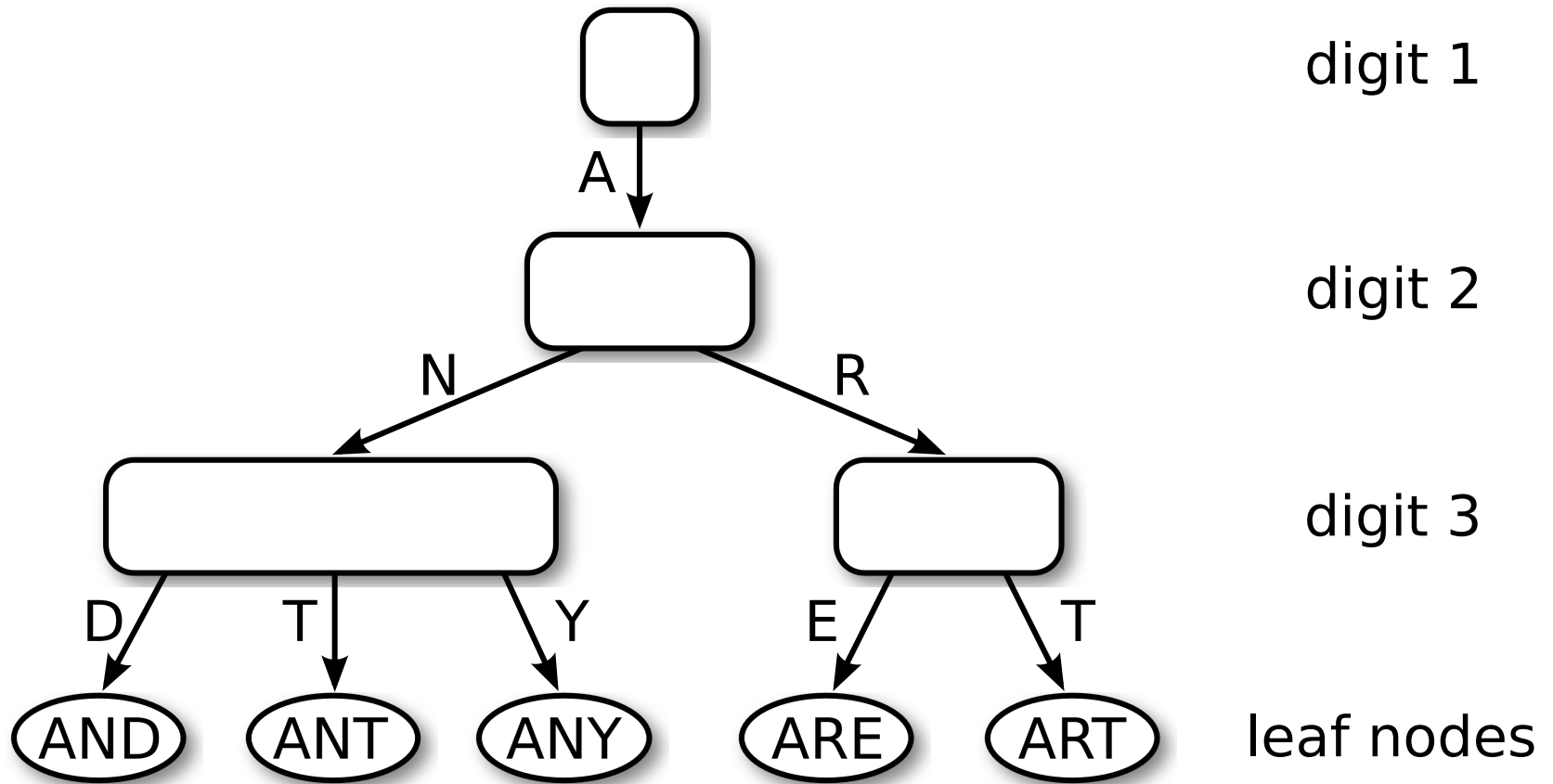


Hauptspeicher-Datenbanken



Hauptspeicher-Datenbanken

ART-Tree (Adaptiver Radix-Baum)





Hauptspeicher-Datenbanken

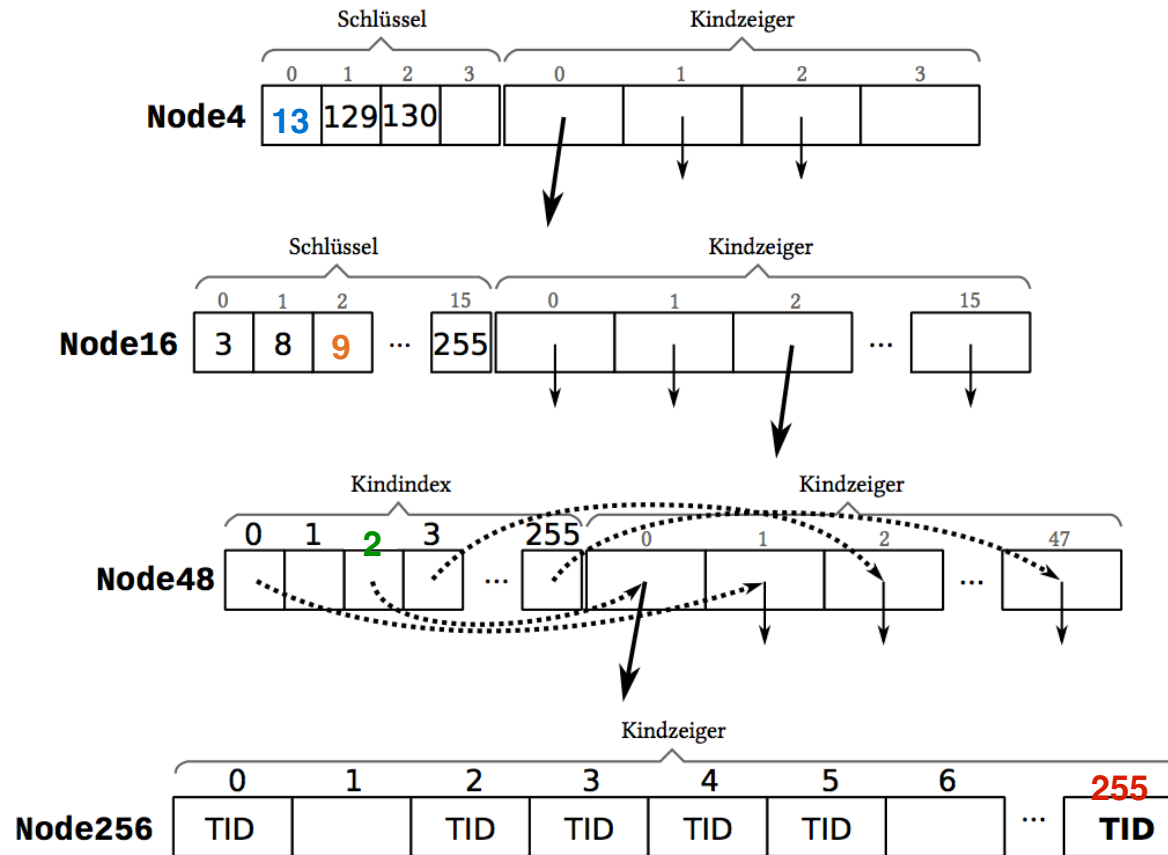
ART-Knotentypen

Integer Schlüssel
+218694399

Bit Repräsentation (32 bit, ohne Vorzeichen)
00001101 00001001 00000010 11111111

Byte-Repräsentation

13	9	2	255
----	---	---	-----





Aufgabe 1

In (pseudo) C++ kann eine ‘Row-Store-artige’ Datenstruktur wie folgt angelegt werden:

```
struct Tuple {  
    int MatrNr;  
    RuntimeString Name;  
    int Semester;  
}  
Tuple data[10000] = {};
```

Notieren Sie, wie die Daten in Form eines Column Stores gehalten werden können in (pseudo) C++.

Erklären Sie Ihrem Tutor, welche Vor- und Nachteile Row- und Column Stores jeweils haben. Was würden Sie für Amazons Webseite verwenden? Was verwenden Sie für die Controlling Datenbank?



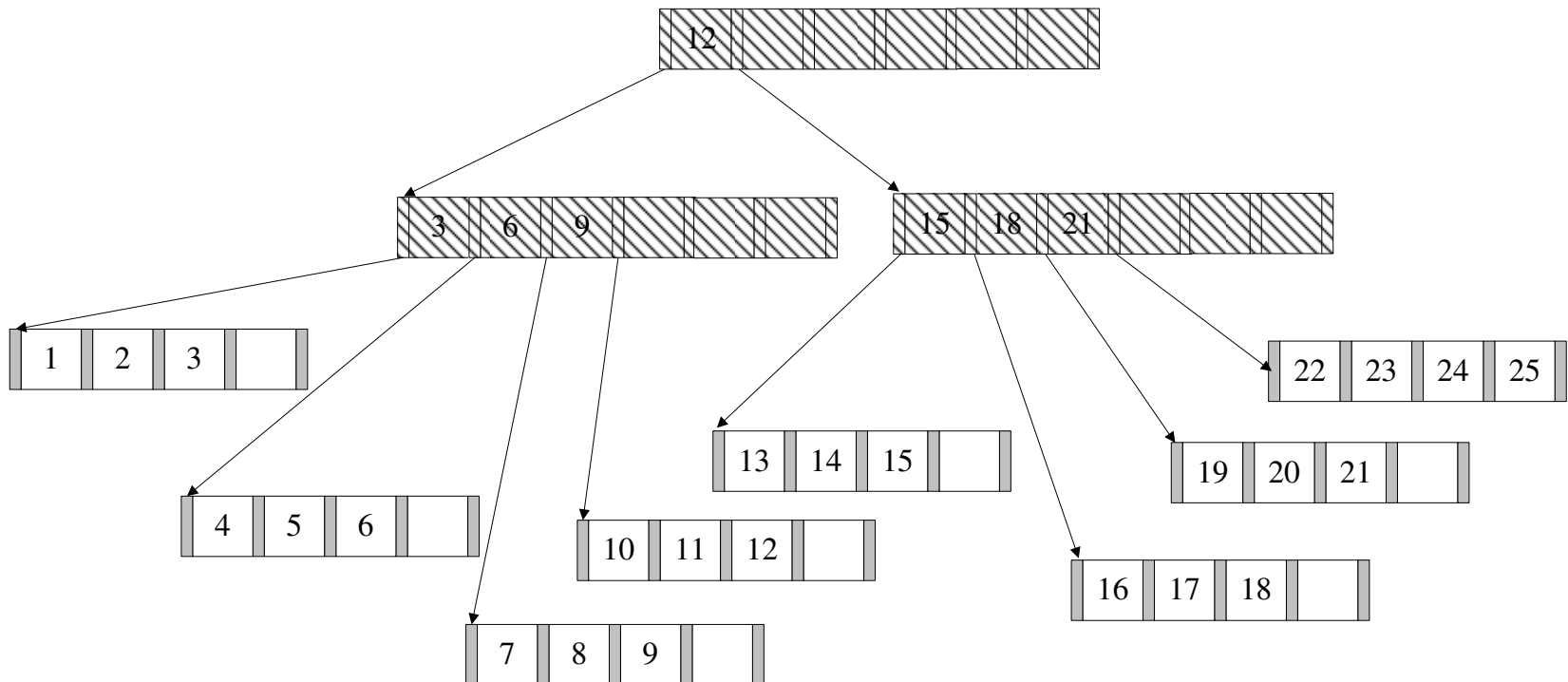
Aufgabe 2

Schätzen sie die Anzahl der Cache Misses die entstehen, wenn man 1001 32-bit Integer Werte (0-1000) in aufeinanderfolgender Reihenfolge in einen ART Baum einfügt. Wäre ein B+ Baum besser oder schlechter? Bei den Baumknoten müssen die Header nicht berücksichtigt werden, Pointer habe eine Größe von 64 bit.



Aufgabe 2

B+ Baum Beispiel



Aufgabe 3

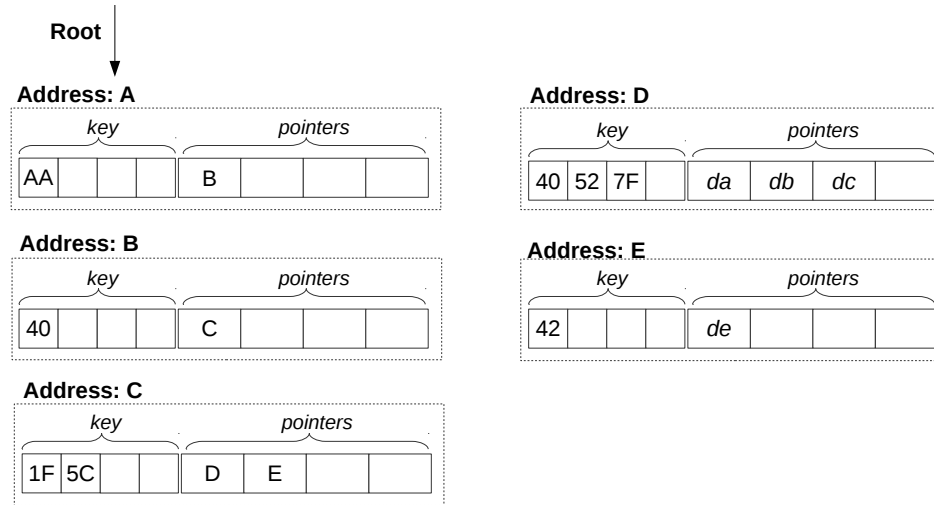


Abbildung 1: Knoten des ART (jeweils Node4)

In Abbildung 1 sehen Sie die Knoten eines ART Baums. Der Wurzelknoten liegt an Adresse A. Zeiger die mit **d** anfangen (z.B. da, db, ...) zeigen auf Daten. Suchschlüssel sind in den Aufgaben jeweils sowohl als Zahl z.B. 99, als auch hexadezimal codiert angegeben, z.B. der Wert 99 als 32 Bit Integer (0x00 0x00 0x00 0x63).

- 1) Beschreiben Sie kurz den Pfad durch den Baum für den 32-bit Suchschlüssel 2856344642 (0xAA 0x40 0x5C 0x42).
- 2) Welche dieser Suchschlüssel sind im Baum enthalten? 291 (0x00 0x00 0x01 0x23), 2856329024 (0xAA 0x40 0x1F 0x40), 2856329026 (0xAA 0x40 0x1F 0x42)
- 3) Beschreiben Sie kurz wie sich der Baum beim Einfügen des Schlüssels 2856352578 (0xAA 0x40 0x7B 0x42) verändert. Der Schlüssel soll auf den Wert an der Adresse **df** zeigen.



Gruppenaufgabe 4

In traditionellen Datenbanksystemen sind die Festplatte und der Buffermanager oft der Hauptgrund für Performanceengpässe. Wie ändert sich dies in Hauptspeicherdatenbanken, wo sind die neuen Flaschenhälse? Unterscheiden Sie auch zwischen Analytischen und Transaktionalen Workloads.



XML-Anfragesprachen

XML (eXtensible Markup Language)

Daten in Baumstruktur und Attributen gespeichert

Schema kann aber muss nicht definiert werden

Basis von HTML

```
<Student ID="M1337" MatrNr="M1337">  
  <Name>1337</Name>  
  <Semester>9</Semester>  
  <hoert Vorlesungen="V5043 V5052 V5259 V5216 V4630"/>  
</Student>
```



XML-Anfragesprachen

XML (eXtensible Markup Language)

Finde die Fehler

```
<Uni Name="Alexander Maximilian Universität" Kuerzel=AMU>
  <UniLeitung>
    <Rektor>Max</Rektor>
    <Senatsvorsitzender>Alex</Senat>
  </UniLeitung>
  <Studenten />
  <Student>
    <Name Peter Name>
    <MatrNr>03670815</MatrNr>
    <Vorlesungen>V1<V2<V3</Vorlesungen>
  </Studenten>
</Uni>
```



XML-Anfragesprachen

XML (eXtensible Markup Language)

Finde die Fehler

Attributwert nicht in Anführungszeichen

```
<Uni Name="Alexander Maximilian Universität" Kuerzel=AMU>
  <UniLeitung>
    <Rektor>Max</Rektor>   Öffnender und schließender Tag ungleich
    <Senatsvorsitzender>Alex</Senat>
  </UniLeitung>
  <Studenten />           Signalisiert leeren Tag, hat aber Inhalt
  <Student>               Schließender Tag fehlt
    <Name Peter Name>     Einfach falsch
    <MatrNr>03670815</MatrNr>
    <Vorlesungen>V1<V2<V3</Vorlesungen>
  </Studenten>
</Uni>
```

< und > dürfen nicht einfach im Text vorkommen



XML-Anfragesprachen

XML (eXtensible Markup Language)

Finde die Fehler

```
<Uni Name="Alexander Maximilian Universität" Kuerzel="AMU">
  <UniLeitung>
    <Rektor>Max</Rektor>
    <Senatsvorsitzender>Alex</Senatsvorsitzender>
  </UniLeitung>
  <Studenten>
    <Student>
      <Name>Peter</Name>
      <MatrNr>03670815</MatrNr>
      <Vorlesungen>V1, V2, V3</Vorlesungen>
    </Student>
  </Studenten>
</Uni>
```



XML-Anfragesprachen

XPath (Finden von Knoten in XML)

Finden von bestimmten Knoten (und allen Nachfahren) im Dokument

Verschiedene Suchachsen zur Navigation durch den Baum

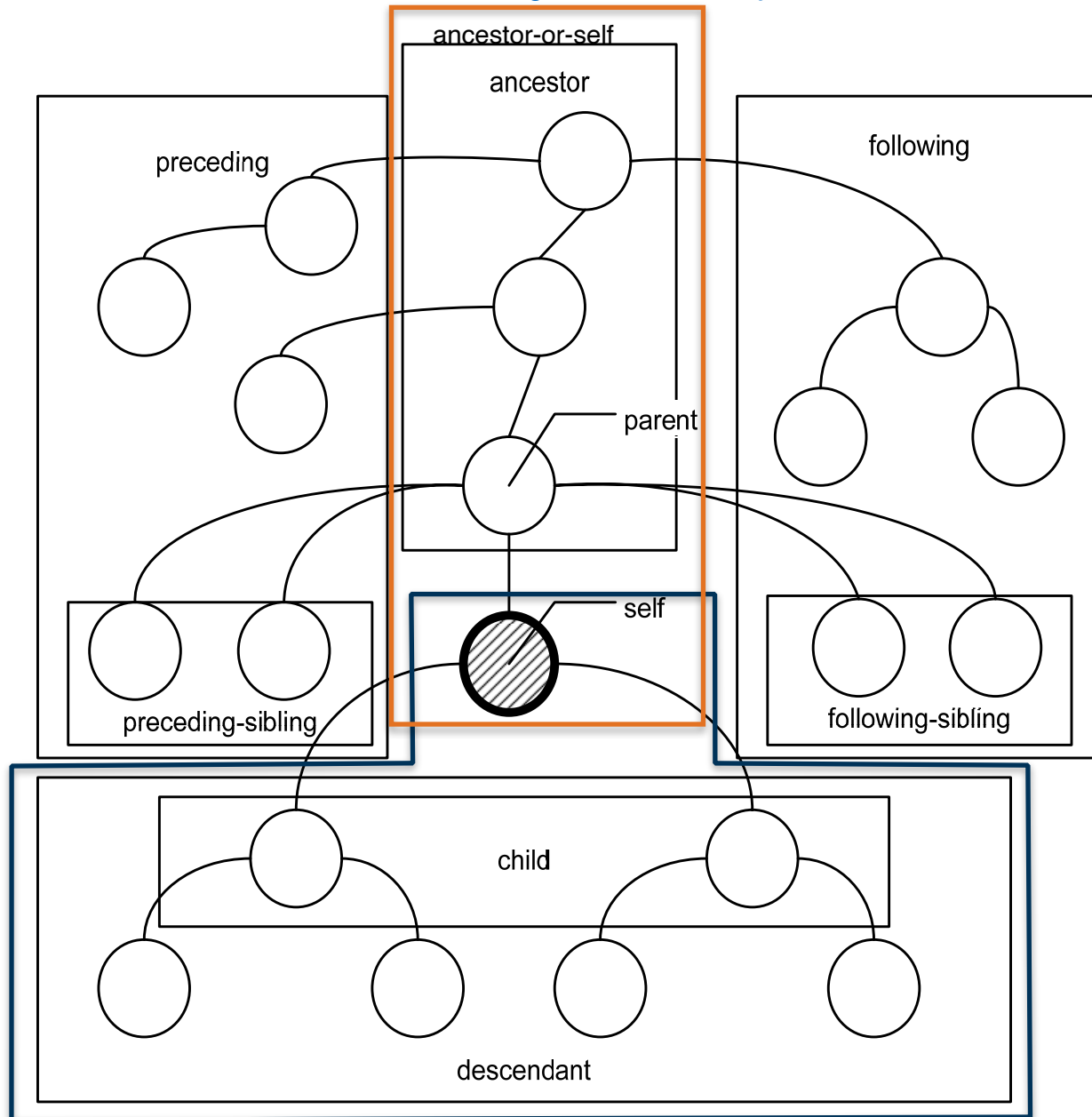
Ausgabe ist Text oder XML-Knoten

Syntax: /Achse::Knotentest[Prädikat]

beliebig oft hintereinander im Ausdruck nutzbar



XML Achsen





XML-Anfragesprachen

Vereinfachte Syntax der XPath-Achsen

<i>Achse</i>	<i>Beschreibung</i>	<i>Abkürzung</i>
<i>self</i>	der Kontextknoten selbst	. (Punkt)
<i>child</i>	direkt untergeordnete Knoten (Kinder)	/
<i>parent</i>	der direkt übergeordnete Elternknoten	.. (2x Punkt)
<i>descendant</i>	untergeordnete Knoten (Nachfahren)	//
<i>attribute</i>	Attributknoten	/@



XML-Anfragesprachen

XPath-Prädikate

Eine beliebige Anzahl kann hintereinander gestellt werden

XPath-Ausdrücke + Funktionen dürfen enthalten sein:

- Knotenindex [i] => i-ter Knoten (Zählung beginnt mit 1)
- Arithmetische Operationen (+, -, *, /, mod)
- Vergleiche (<, >, <=, >=, !=, =)
- logische Operationen (and, or)
- Aggregatsfunktionen (min, max, count)

```
doc('uni2')//ProfessorIn[count(../Vorlesung)=3]
```



XML-Anfragesprachen

XPath-Prädikate

```
<Uni Name="Alexander Maximilian Universität" Kuerzel="AMU">
  <UniLeitung>
    <Rektor>Max</Rektor>
    <Senatsvorsitzender>Alex</Senatsvorsitzender>
  </UniLeitung>
  <Studenten>
    <Student>
      <Name>Peter</Name>
      <MatrNr>03670815</MatrNr>
      <Vorlesungen>V1,V2,V3</Vorlesungen>
    </Student>
  </Studenten>
</Uni>
```

```
doc('uniDoc')/Uni/Studenten/Student[Name = 'Peter']/MatrNr
```



XML-Anfragesprachen

XPath-Prädikate

```
<Uni Name="Alexander Maximilian Universität" Kuerzel="AMU">
  <UniLeitung>
    <Rektor>Max</Rektor>
    <Senatsvorsitzender>Alex</Senatsvorsitzender>
  </UniLeitung>
  <Studenten>
    <Student>
      <Name>Peter</Name>
      <MatrNr>03670815</MatrNr>
      <Vorlesungen>V1,V2,V3</Vorlesungen>
    </Student>
  </Studenten>
</Uni>
```

`doc('uniDoc')/Uni/Studenten/Student[Name = 'Peter']/MatrNr`

Ausgabe: `<MatrNr>03670815</MatrNr>`



XML-Anfragesprachen

XPath-Prädikate

```
<Uni Name="Alexander Maximilian Universität" Kuerzel="AMU">
  <UniLeitung>
    <Rektor>Max</Rektor>
    <Senatsvorsitzender>Alex</Senatsvorsitzender>
  </UniLeitung>
  <Studenten>
    <Student>
      <Name>Peter</Name>
      <MatrNr>03670815</MatrNr>
      <Vorlesungen>V1,V2,V3</Vorlesungen>
    </Student>
  </Studenten>
</Uni>
```

Element



```
doc('uniDoc')//Student[Name = 'Peter']/MatrNr
```



XML-Anfragesprachen

XPath-Prädikate

```
<Uni Name="Alexander Maximilian Universität" Kuerzel="AMU">
  <UniLeitung>
    <Rektor>Max</Rektor>
    <Senatsvorsitzender>Alex</Senatsvorsitzender>
  </UniLeitung>
  <Studenten>
    <Student>
      <Name>Peter</Name>
      <MatrNr>03670815</MatrNr>
      <Vorlesungen>V1,V2,V3</Vorlesungen>
    </Student>
  </Studenten>
</Uni>
```

Attribut



`doc('uniDoc')/Uni[@Name = 'Alexander Maximilian Universität']/UniLeitung/Rektor`

```
<Rektor>Max</Rektor>
```



XML-Anfragesprachen

XPath-Prädikate

```
<Uni Name="Alexander Maximilian Universität" Kuerzel="AMU">
  <UniLeitung>
    <Rektor>Max</Rektor>
    <Senatsvorsitzender>Alex</Senatsvorsitzender>
  </UniLeitung>
  <Studenten>
    <Student>
      <Name>Peter</Name>
      <MatrNr>03670815</MatrNr>
      <Vorlesungen>V1,V2,V3</Vorlesungen>
    </Student>
  </Studenten>
</Uni>
```

```
doc('uniDoc')//Student[Name = 'Peter']/../@Name
```

```
Name="Alexander Maximilian Universität"
```



Hinweise Die Aufgaben können auf <http://xquery.db.in.tum.de/> getestet werden. Die Daten für das Unischema können mit `doc('uni2')` geladen werden. Zur Lösung der Aufgaben können sie die folgenden XQuery-Funktionen verwenden:

`max(NUM)`, `count(X)`, `tokenize(STR,SEP)`, `sum(NUM)`, `contains(HAY,NEEDLE)`

1. `max(NUMBERS)` - Returns largest number from list
2. `count(LIST)` - Return the number of elements in the list
3. `tokenize(STR,SEP)` - Splits up the string at the separator
4. `sum(NUMBERS)` - Returns sum of all numbers in list
5. `contains(HAY,NEEDLE)` - Checks if the search string (`NEEDLE`) is contained in the string (`HAY`)
6. `distinct-values(LIST)` - Returns the distinct values from the list



Aufgabe 5

Lösen Sie in XPath folgende Aufgaben und testen Sie diese auf `xquery.db.in.tum.de`.

1. Lassen Sie sich das gesamte Schema anzeigen.
2. Finden Sie die Namen aller Fakultäten.
3. Finden Sie die Namen aller Studenten, die Vorlesungen hören.



Aufgabe 5

```
<Universitaet UnivName="Virtuelle Universitaet ...">
  <UniLeitung>... </UniLeitung>
  <Fakultaeten>
    <Fakultaet>
      <FakName>Theologie</FakName>
      <ProfessorIn ID="P2134" PersNr="P2134">
        <Name>Augustinus</Name>
        <Rang>C3</Rang>
        <Raum>309</Raum>
        <Vorlesungen>
          <Vorlesung ID="V5022" VorlNr="V5022">
            <Titel>Glaube und Wissen</Titel>
            <SWS>2</SWS>
          </Vorlesung>
        </Vorlesungen>
        <Assistenten>
          <Assistent ID="P3007" PersNr="P3007">
            <Name>Spinoza</Name>
            <Fachgebiet>Gott und Natur</Fachgebiet>
          </Assistent>
        </Assistenten>
      </ProfessorIn>
    </Fakultaet>
  ...
</Universitaet>
```

```
</Fakultaeten>
<Studenten>
  <Student ID="M24002" MatrNr="M24002">
    <Name>Xenokrates</Name>
    <Semester>18</Semester>
  </Student>
  <Student ID="M25403" MatrNr="M25403">
    <Name>Jonas</Name>
    <Semester>12</Semester>
    <hoert Vorlesungen="V5022"/>
    <Pruefungen>
      <Pruefung Pruefer="P2125" Vorlesung="V5041"
        Note="2.0"/>
    </Pruefungen>
  </Student>
  ...
  <Student ID="M1337" MatrNr="M1337">
    <Name>1337</Name>
    <Semester>9</Semester>
    <hoert Vorlesungen="V5022 V5041 ... V4630"/>
  </Student>
</Studenten>
</Universitaet>
```

1. Lassen Sie sich das gesamte Schema anzeigen.



Aufgabe 5

```
<Universitaet UnivName="Virtuelle Universitaet ...">
  <UniLeitung>... </UniLeitung>
  <Fakultaeten>
    <Fakultaet>
      <FakName>Theologie</FakName>
      <ProfessorIn ID="P2134" PersNr="P2134">
        <Name>Augustinus</Name>
        <Rang>C3</Rang>
        <Raum>309</Raum>
        <Vorlesungen>
          <Vorlesung ID="V5022" VorlNr="V5022">
            <Titel>Glaube und Wissen</Titel>
            <SWS>2</SWS>
          </Vorlesung>
        </Vorlesungen>
        <Assistenten>
          <Assistent ID="P3007" PersNr="P3007">
            <Name>Spinoza</Name>
            <Fachgebiet>Gott und Natur</Fachgebiet>
          </Assistent>
        </Assistenten>
      </ProfessorIn>
    </Fakultaet>
    ...
  </Fakultaeten>
</Universitaet>
```

```
</Fakultaeten>
<Studenten>
  <Student ID="M24002" MatrNr="M24002">
    <Name>Xenokrates</Name>
    <Semester>18</Semester>
  </Student>
  <Student ID="M25403" MatrNr="M25403">
    <Name>Jonas</Name>
    <Semester>12</Semester>
    <hoert Vorlesungen="V5022"/>
    <Pruefungen>
      <Pruefung Pruefer="P2125" Vorlesung="V5041"
        Note="2.0"/>
    </Pruefungen>
  </Student>
  ...
  <Student ID="M1337" MatrNr="M1337">
    <Name>1337</Name>
    <Semester>9</Semester>
    <hoert Vorlesungen="V5022 V5041 ... V4630"/>
  </Student>
</Studenten>
</Universitaet>
```

2. Finden Sie die Namen aller Fakultäten.



Aufgabe 5

```
<Universitaet UnivName="Virtuelle Universitaet ...">
  <UniLeitung>... </UniLeitung>
  <Fakultaeten>
    <Fakultaet>
      <FakName>Theologie</FakName>
      <ProfessorIn ID="P2134" PersNr="P2134">
        <Name>Augustinus</Name>
        <Rang>C3</Rang>
        <Raum>309</Raum>
        <Vorlesungen>
          <Vorlesung ID="V5022" VorlNr="V5022">
            <Titel>Glaube und Wissen</Titel>
            <SWS>2</SWS>
          </Vorlesung>
        </Vorlesungen>
        <Assistenten>
          <Assistent ID="P3007" PersNr="P3007">
            <Name>Spinoza</Name>
            <Fachgebiet>Gott und Natur</Fachgebiet>
          </Assistent>
        </Assistenten>
      </ProfessorIn>
    </Fakultaet>
  ...
```

```
</Fakultaeten>
<Studenten>
  <Student ID="M24002" MatrNr="M24002">
    <Name>Xenokrates</Name>
    <Semester>18</Semester>
  </Student>
  <Student ID="M25403" MatrNr="M25403">
    <Name>Jonas</Name>
    <Semester>12</Semester>
    <hoert Vorlesungen="V5022"/>
    <Pruefungen>
      <Pruefung Pruefer="P2125" Vorlesung="V5041"
        Note="2.0"/>
    </Pruefungen>
  </Student>
  ...
  <Student ID="M1337" MatrNr="M1337">
    <Name>1337</Name>
    <Semester>9</Semester>
    <hoert Vorlesungen="V5022 V5041 ... V4630"/>
  </Student>
</Studenten>
</Universitaet>
```

3. Finden Sie die Namen aller Studenten, die Vorlesungen hören.



XML-Anfragesprachen (XQuery)



XML-Anfragesprachen

XQuery

Basiert auf XPath und kombiniert Ergebnisse der Anfragen

FLWOR-Syntax

For Schleifen

Let Variablen definieren

Where Selektieren

Order By Sortieren

Return Ergebnis als neues XML formatieren



XML-Anfragesprachen

XQuery

Es muss nicht die komplette FLWOR Syntax genutzt werden, aber immer wenn FLW oder O genutzt werden, braucht man return

Variablen dürfen XML oder Unterabfragen (XPath oder XQuery) enthalten
Alle Variablen beginnen mit \$

Beim Einbetten von XQuery in XML müssen geschweifte Klammern benutzt werden (und auch nur dann)

```
<XML>{XQuery}</XML>
```



XML-Anfragesprachen

XQuery-Beispielsanfrage

```
<Professoren>
{
  for $p in doc('uni2')//ProfessorIn
    let $v := $p/Vorlesungen/Vorlesung
    where count ($v) > 0
    order by sum ($v/SWS)
    return
      <ProfessorIn>
        {$p/Name}
        <Belastung>{sum($v/SWS)}</Belastung>
      </ProfessorIn>
}
</Professoren>
```




XML-Anfragesprachen

XQuery Ergebnis

```
<Professoren>
{
  for $p in doc('uni2')//ProfessorIn
  let $v := $p/Vorlesungen/Vorlesung
  where count ($v) > 0
  order by sum ($v/SWS)
  return
    <ProfessorIn>
      {$p/Name}
      <Belastung>{sum($v/SWS)}</Belastung>
    </ProfessorIn>
}
</Professoren>
```

Ausgabe:

```
<Professoren>
  <ProfessorIn>
    <Name>Russel</Name>
    <Belastung>6</Belastung>
  </ProfessorIn>
  <ProfessorIn>
    <Name>Sokrates</Name>
    <Belastung>10</Belastung>
  </ProfessorIn>
  <ProfessorIn>
    <Name>Kant</Name>
    <Belastung>10</Belastung>
  </ProfessorIn>
</Professoren>
```



Aufgabe 6

Lösen Sie mit XQuery folgende Anfragen und testen Sie diese auf `xquery.db.in.tum.de`.

1. Geben Sie eine nach Rang sortierte Liste der Professoren aus (C4 oben).
2. Finden Sie die Namen der Professoren, die die meisten Assistenten haben.
3. Finden Sie für jede von einem Student gehörte Prüfung den Namen des Prüfers und Vorlesung.



Aufgabe 6

```
<Universitaet UnivName="Virtuelle Universitaet ...">
  <UniLeitung>... </UniLeitung>
  <Fakultaeten>
    <Fakultaet>
      <FakName>Theologie</FakName>
      <ProfessorIn ID="P2134" PersNr="P2134">
        <Name>Augustinus</Name>
        <Rang>C3</Rang>
        <Raum>309</Raum>
        <Vorlesungen>
          <Vorlesung ID="V5022" VorlNr="V5022">
            <Titel>Glaube und Wissen</Titel>
            <SWS>2</SWS>
          </Vorlesung>
        </Vorlesungen>
        <Assistenten>
          <Assistent ID="P3007" PersNr="P3007">
            <Name>Spinoza</Name>
            <Fachgebiet>Gott und Natur</Fachgebiet>
          </Assistent>
        </Assistenten>
      </ProfessorIn>
    </Fakultaet>
    ...
  </Fakultaeten>
```

```
<Studenten>
  <Student ID="M24002" MatrNr="M24002">
    <Name>Xenokrates</Name>
    <Semester>18</Semester>
  </Student>
  <Student ID="M25403" MatrNr="M25403">
    <Name>Jonas</Name>
    <Semester>12</Semester>
    <hoert Vorlesungen="V5022"/>
    <Pruefungen>
      <Pruefung Pruefer="P2125" Vorlesung="V5041"
        Note="2.0"/>
    </Pruefungen>
  </Student>
  ...
  <Student ID="M1337" MatrNr="M1337">
    <Name>1337</Name>
    <Semester>9</Semester>
    <hoert Vorlesungen="V5022 V5041 ... V4630"/>
  </Student>
</Studenten>
</Universitaet>
```

1. Geben Sie eine nach Rang sortierte Liste der Professoren aus (C4 oben).



Aufgabe 6

```
<Universitaet UnivName="Virtuelle Universitaet ...">
  <UniLeitung>... </UniLeitung>
  <Fakultaeten>
    <Fakultaet>
      <FakName>Theologie</FakName>
      <ProfessorIn ID="P2134" PersNr="P2134">
        <Name>Augustinus</Name>
        <Rang>C3</Rang>
        <Raum>309</Raum>
        <Vorlesungen>
          <Vorlesung ID="V5022" VorlNr="V5022">
            <Titel>Glaube und Wissen</Titel>
            <SWS>2</SWS>
          </Vorlesung>
        </Vorlesungen>
        <Assistenten>
          <Assistent ID="P3007" PersNr="P3007">
            <Name>Spinoza</Name>
            <Fachgebiet>Gott und Natur</Fachgebiet>
          </Assistent>
        </Assistenten>
      </ProfessorIn>
    </Fakultaet>
    ...
  </Fakultaeten>
```

```
<Studenten>
  <Student ID="M24002" MatrNr="M24002">
    <Name>Xenokrates</Name>
    <Semester>18</Semester>
  </Student>
  <Student ID="M25403" MatrNr="M25403">
    <Name>Jonas</Name>
    <Semester>12</Semester>
    <hoert Vorlesungen="V5022"/>
    <Pruefungen>
      <Pruefung Pruefer="P2125" Vorlesung="V5041"
        Note="2.0"/>
    </Pruefungen>
  </Student>
  ...
  <Student ID="M1337" MatrNr="M1337">
    <Name>1337</Name>
    <Semester>9</Semester>
    <hoert Vorlesungen="V5022 V5041 ... V4630"/>
  </Student>
</Studenten>
</Universitaet>
```

2. Finden Sie die Namen der Professoren, die die meisten Assistenten haben.



Aufgabe 6

```
<Universitaet UnivName="Virtuelle Universitaet ...">
  <UniLeitung>... </UniLeitung>
  <Fakultaeten>
    <Fakultaet>
      <FakName>Theologie</FakName>
      <ProfessorIn ID="P2134" PersNr="P2134">
        <Name>Augustinus</Name>
        <Rang>C3</Rang>
        <Raum>309</Raum>
        <Vorlesungen>
          <Vorlesung ID="V5022" VorlNr="V5022">
            <Titel>Glaube und Wissen</Titel>
            <SWS>2</SWS>
          </Vorlesung>
        </Vorlesungen>
        <Assistenten>
          <Assistent ID="P3007" PersNr="P3007">
            <Name>Spinoza</Name>
            <Fachgebiet>Gott und Natur</Fachgebiet>
          </Assistent>
        </Assistenten>
      </ProfessorIn>
    </Fakultaet>
    ...
  </Fakultaeten>
```

```
<Studenten>
  <Student ID="M24002" MatrNr="M24002">
    <Name>Xenokrates</Name>
    <Semester>18</Semester>
  </Student>
  <Student ID="M25403" MatrNr="M25403">
    <Name>Jonas</Name>
    <Semester>12</Semester>
    <hoert Vorlesungen="V5022"/>
    <Pruefungen>
      <Pruefung Pruefer="P2125" Vorlesung="V5041"
        Note="2.0"/>
    </Pruefungen>
  </Student>
  ...
  <Student ID="M1337" MatrNr="M1337">
    <Name>1337</Name>
    <Semester>9</Semester>
    <hoert Vorlesungen="V5022 V5041 ... V4630"/>
  </Student>
</Studenten>
</Universitaet>
```

3. Finden Sie für jede von einem Student gehörte Prüfung den Namen des Prüfers und Vorlesung.



Fragen?