



Einsatz und Realisierung von Datenbanksystemen

ERDB Übungsleitung

Alice Rey, Maximilian Reif, Tobias Götz

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.

MVCC

- **Multiversion Concurrency Control**
- fein granulares Mehrversionsverfahren für die Synchronisation mehrerer paralleler TA
- sehr gut für unterbrechbare TA
- klassische DB nutzen das 2-Phasen-Sperrprotokoll
zu schwerfällig für Hauptspeicher DBs



MVCC

Version positions
in a fixed range
[first,last)

Accounts

VersionVector

Owner	Balance	VersionVector
Thomas	10	
Larry	10	
Alfons	10	
Judy	10	
Tobias	10	
Sally	10	
Hanna	10	
Hasso	10	
Mike	10	
Lisa	10	
Betty	10	
Cindy	10	
Henry	10	
Praveen	10	
Wendy	10	

base table (column-store)

(not stored – for illustration)
Actions
commitTime

recently committed

startTime
Tx-ID
Actions
(not stored)

active Transactions



MVCC

Version positions
in a fixed range
[first,last)

Accounts

Owner	Balance	VersionVector
Thomas	10	
Larry	10	
Alfons	10	
Judy	10	
Tobias	10	
Sally	9	
Hanna	10	
Hasso	10	
Mike	10	
Lisa	10	
Betty	10	
Cindy	10	
Henry	10	
Praveen	10	
Wendy	10	

base table (column-store)

VersionVector

latest version

in-place

physical before images (i.e. column
values) in undo buffers

Undo buffer of Ta

Ta, Balance, 10

(not stored – for illustration)
Actions
commitTime

recently committed

startTime
Tx-ID
Actions
(not stored)

Ta	T1	Sally → Wen

active Transactions



MVCC

Version positions
in a fixed range
[first,last)

Accounts

Owner	Balance	VersionVector
Thomas	10	
Larry	10	
Alfons	10	
Judy	10	
Tobias	10	
Sally	9	
Hanna	10	
Hasso	10	
Mike	10	
Lisa	10	
Betty	10	
Cindy	10	
Henry	10	
Praveen	10	
Wendy	11	

base table (column-store)

VersionVector

latest version
in-place

Undo buffer of Ta

Ta,Balance,10 Ta,Balance,10

(not stored – for illustration)
Actions
commitTime

recently committed

startTime
Tx-ID
Actions
(not stored)

Ta	T1	Sally → Wen

active Transactions



MVCC

Version positions
in a fixed range
[first,last)

Accounts

Owner	Balance	VersionVector
Thomas	10	
Larry	10	
Alfons	10	
Judy	10	
Tobias	10	
Sally	9	
Hanna	10	
Hasso	10	
Mike	10	
Lisa	10	
Betty	10	
Cindy	10	
Henry	10	
Praveen	10	
Wendy	11	

base table (column-store)

VersionVector

latest version
in-place

Try update of uncommitted

Undo buffer of Tb

Tb,Balance,9

Undo buffer of Ta

Ta,Balance,10

Ta,Balance,10

(not stored – for illustration)
Actions
commitTime

recently committed

startTime
Tx-ID
Actions
(not stored)

Ta	T1	Sally → Wen
Tb	T2	Sally → Hen

active Transactions



MVCC

Version positions
in a fixed range
[first,last)

Accounts

Owner	Balance	VersionVector
Thomas	10	
Larry	10	
Alfons	10	
Judy	10	
Tobias	10	
Sally	9	
Hanna	10	
Hasso	10	
Mike	10	
Lisa	10	
Betty	10	
Cindy	10	
Henry	10	
Praveen	10	
Wendy	11	

base table (column-store)

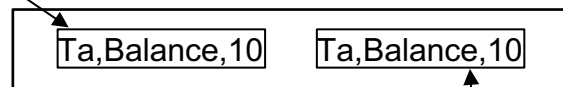
VersionVector

latest version
in-place

Restart the transaction
Undo buffer of Tb



Undo buffer of Ta



(not stored – for illustration)
Actions
commitTime

recently committed

startTime
Tx-ID
Actions
(not stored)

Ta	T1	Sally → Wen
Tb		Sally → Hen

active Transactions



MVCC

Version positions
in a fixed range
[first,last)

Accounts

Owner	Balance	VersionVector
Thomas	10	
Larry	10	
Alfons	10	
Judy	10	
Tobias	10	
Sally	9	
Hanna	10	
Hasso	10	
Mike	10	
Lisa	10	
Betty	10	
Cindy	10	
Henry	10	
Praveen	10	
Wendy	11	

base table (column-store)

VersionVector

latest version
in-place

Undo buffer of Tb

Commit transaction

Undo buffer of T3

T3,Balance,10

T3,Balance,10

(not stored – for illustration)
Actions
commitTime

T3	Sally → Wendy

recently committed

startTime
Tx-ID
Actions
(not stored)

Tb		Sally → Hen
Tb		Sally → Hen

active Transactions

MVCC

Version positions
in a fixed range
[first,last)

Accounts

Owner	Balance	VersionVector
Thomas	10	
Larry	10	
Alfons	10	
Judy	10	
Tobias	10	
Sally	8	
Hanna	10	
Hasso	10	
Mike	10	
Lisa	10	
Betty	10	
Cindy	10	
Henry	10	
Praveen	10	
Wendy	11	

base table (column-store)

VersionVector

latest version
in-place

physical before images (i.e. column
values) in undo buffers

Undo buffer of Tb

Tb,Balance,9

Undo buffer of T3

T3,Balance,10

T3,Balance,10

(not stored – for illustration)
commitTime
Actions

T3	Sally → Wendy

recently committed

startTime
Tx-ID
Actions
(not stored)

Tb	T4	Sally → Hen

active Transactions

MVCC

Version positions
in a fixed range
[first,last)

Accounts

Owner	Balance	VersionVector
Thomas	10	
Larry	10	
Alfons	10	
Judy	10	
Tobias	10	
Sally	8	
Hanna	10	
Hasso	10	
Mike	10	
Lisa	10	
Betty	10	
Cindy	10	
Henry	11	
Praveen	10	
Wendy	11	

base table (column-store)

VersionVector

latest version
in-place

Undo buffer of Tb

Tb,Balance,9 Tb,Balance,10

Undo buffer of T3

T3,Balance,10 T3,Balance,10

(not stored – for illustration)
commitTime Actions

T3	Sally → Wendy

recently committed

startTime Tx-ID Actions
(not stored)

Tb	T4	Sally → Hen

active Transactions

MVCC

Version positions
in a fixed range
[first,last)

Accounts

Owner	Balance	VersionVector
Thomas	10	
Larry	10	
Alfons	10	
Judy	10	
Tobias	10	
Sally	8	
Hanna	10	
Hasso	10	
Mike	10	
Lisa	10	
Betty	10	
Cindy	10	
Henry	11	
Praveen	10	
Wendy	11	

base table (column-store)

VersionVector

latest version
in-place

Undo buffer of Tb

Tb,Balance,9 Tb,Balance,10

Undo buffer of T3

T3,Balance,10 T3,Balance,10

(not stored – for illustration)
Actions
commitTime

T3	Sally → Wendy

recently committed

(not stored)
Actions
startTime
Tx-ID

Tb	T4	Sally → Hen
Tx	T5	Leser: Σ

active Transactions

MVCC

Version positions
in a fixed range
[first,last)

Accounts

Owner	Balance	VersionVector
Thomas	10	
Larry	10	
Alfons	10	
Judy	10	
Tobias	10	
Sally	8	
Hanna	10	
Hasso	10	
Mike	10	
Lisa	10	
Betty	10	
Cindy	10	
Henry	11	
Praveen	10	
Wendy	11	

base table (column-store)

VersionVector

latest version
in-place

Undo buffer of T6

T6,Balance,9 T6,Balance,10

Undo buffer of T3

T3,Balance,10 T3,Balance,10

(not stored – for illustration)
commitTime Actions

T3	Sally → Wendy
T6	Sally → Henry

recently committed

startTime Tx-ID Actions
(not stored)

Tx	T5	Leser: Σ

active Transactions

MVCC

Version positions
in a fixed range
[first,last)

Accounts

Owner	Balance	VersionVector
Thomas	10	
Larry	10	
Alfons	10	
Judy	10	
Tobias	10	
Sally	7	
Hanna	10	
Hasso	10	
Mike	10	
Lisa	10	
Betty	10	
Cindy	10	
Henry	11	
Praveen	10	
Wendy	11	

base table (column-store)

VersionVector

latest version
in-place

physical before images (i.e. column
values) in undo buffers

Undo buffer of Ty

Ty,Balance,8

Undo buffer of T6

T6,Balance,9

T6,Balance,10

Undo buffer of T3

T3,Balance,10

T3,Balance,10

(not stored – for illustration)
commitTime
Actions

T3	Sally → Wendy
T6	Sally → Henry

recently committed

startTime
Tx-ID
Actions
(not stored)

Tx	T5	Leser: Σ
Ty	T7	Sally → Mike

active Transactions

MVCC

Version positions
in a fixed range
[first,last)

Accounts

Owner	Balance	VersionVector
Thomas	10	
Larry	10	
Alfons	10	
Judy	10	
Tobias	10	
Sally	7	
Hanna	10	
Hasso	10	
Mike	11	
Lisa	10	
Betty	10	
Cindy	10	
Henry	11	
Praveen	10	
Wendy	11	

base table (column-store)

VersionVector

latest version
in-place

physical before images (i.e. column
values) in undo buffers

Undo buffer of Ty

Ty, Balance, 8	Ty, Balance, 10
----------------	-----------------

Undo buffer of T6

T6, Balance, 9	T6, Balance, 10
----------------	-----------------

Undo buffer of T3

T3, Balance, 10	T3, Balance, 10
-----------------	-----------------

(not stored – for illustration)
commitTime
Actions

T3	Sally → Wendy
T6	Sally → Henry

recently committed

startTime
Tx-ID
Actions
(not stored)

Tx	T5	Leser: Σ
	Ty	T7

active Transactions

MVCC

Version positions
in a fixed range
[first,last)

Accounts

Owner	Balance	VersionVector
Thomas	10	
Larry	10	
Alfons	10	
Judy	10	
Tobias	10	
Sally	7	
Hanna	10	
Hasso	10	
Mike	11	
Lisa	10	
Betty	10	
Cindy	10	
Henry	11	
Praveen	10	
Wendy	11	

base table (column-store)

VersionVector

latest version
in-place

physical before images (i.e. column
values) in undo buffers

Undo buffer of Ty

Ty, Balance, 8	Ty, Balance, 10
----------------	-----------------

Undo buffer of T6

T6, Balance, 9	T6, Balance, 10
----------------	-----------------

Undo buffer of T3

T3, Balance, 10	T3, Balance, 10
-----------------	-----------------

(not stored – for illustration)
commitTime
Actions

T3	Sally → Wendy
T6	Sally → Henry

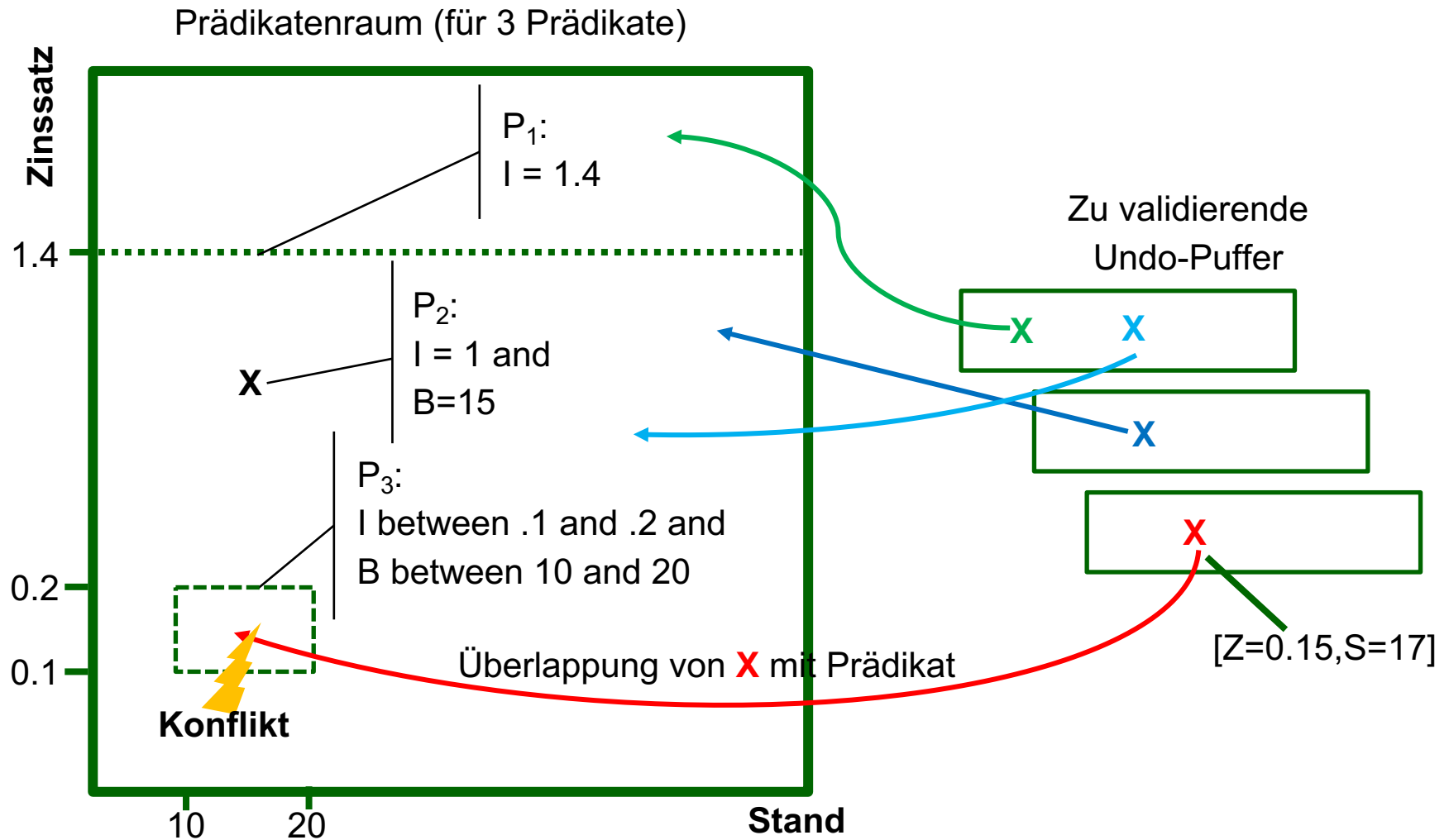
recently committed

startTime
Tx-ID
Actions
(not stored)

Tx	T5	Leser: Σ
	Ty	T7
	Tz	T8
		Leser: Σ

active Transactions

Precision Locking



MVCC mit Precision Locking

- Lesende Anfragen sind **immer erlaubt**: kein Precision Locking
- Falls schreibende Anfrage: **Überlappender Prädikatbereich?**
- Falls ja, dann **BOT** und **commit-Reihenfolge** beachten

Aufgabe 1

T1: insert into foo (select Note from Noten where MatrNr=12345)

T2: insert into bar (select count(*) from Noten where Note<1.5)

T3: insert into Noten(MatNr,Note) values (54321, 3.0)

T4: update Noten set Note=1.4 where MatrNr=32154

T5: insert into Noten(MatNr,Note) values (54321, 1.3)

T6: update Noten set Note=1.6 where MatrNr=12345

Analysieren Sie, ob die folgenden Historien unter dem MVCC Model, wie in der Vorlesung vorgestellt, auftreten können. Jede Historie steht für sich selbst und startet jeweils von einem ursprünglichen Datenzustand. Die Buchstaben innerhalb der Klammer entsprechen dabei jeweils den Tupeln auf die zugegriffen wird. Wenn in T2 z.B. drei Werte das 'Prädikat Note<1.5' erfüllen, gäbe es entsprechend drei $r(\dots)$ Einträge auf die jeweiligen Tupel.

H1 (T1 und T3): $bot_1, r_1(A), bot_3, w_3(B), w_1(C), commit_1, commit_3$

H2 (T2 und T3): $bot_2, r_2(A), bot_3, w_3(B), r_2(C), w_2(D), commit_2, commit_3$

H8 (T2 und T3): $bot_2, r_2(A), bot_3, w_3(B), r_2(C), commit_3, w_2(D), commit_2$

H3 (T2 und T4): $bot_2, r_2(A), r_2(B), bot_4, r_4(B), w_4(B), r_2(C), w_2(D), commit_2, commit_4$

H5 (T2 und T4): $bot_2, r_2(A), bot_4, r_4(B), w_4(B), r_2(C), commit_4, w_2(D), commit_2$

H4 (T1 und T6): $bot_1, r_1(B), bot_6, r_6(B), w_6(B), w_1(C), commit_1, commit_6$

H6 (T1 und T6): $bot_1, r_1(B), bot_6, r_6(B), w_6(B), commit_6, w_1(C), commit_1$

H7 (T2 und T5): $bot_2, r_2(A), bot_5, w_5(D), commit_5, r_2(D), w_2(E), commit_2$

H9 (T2 und T5): $bot_2, r_2(A), bot_5, w_5(B), r_2(C), commit_5, w_2(D), commit_2$

Aufgabe 2

Name	verfügbar	Versionsvektor	TID	Startzeit	Commitzeit	Aktion
House	ja	-	Ta	$T0$	-	$\sum$
Green	ja	-	Tb	$T2$	-	$(Green) - -$
Brinkmann	ja	-	Tc	$T3$	-	$\sum$
			Td	$T5$	-	$\sum$

Beschäftigen wir uns mit *Multi-Version Concurrency Control* am Beispiel unserer verfügbaren Ärzte („Doctors on call/duty“), in dem wir sicherstellen wollen, dass immer mindestens ein Arzt verfügbar ist.

Uns stehen drei Operationen zur Verfügung, $\sum$ zählt alle verfügbaren Ärzte, $(X) +$ ändert Xs Status in verfügbar, $(X) - -$ zählt alle verfügbaren Ärzte und ändert Xs Status auf nicht verfügbar, wenn mindestens ein Arzt noch anwesend ist.

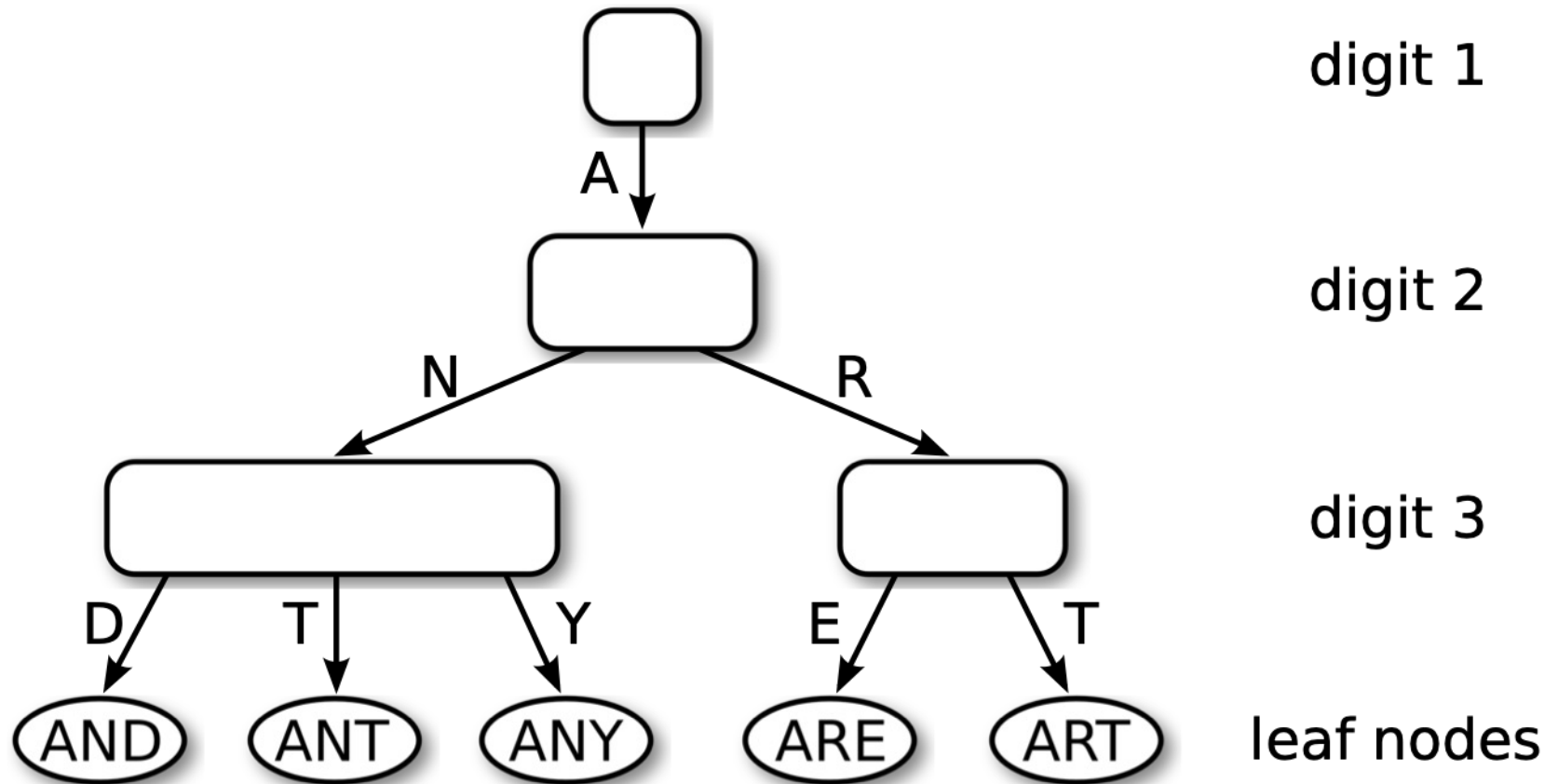
1. Welche Bedingungen gelten für die Zeitstempel?
2. Green möchte zum Zeitpunkt $T2$ seinen Feierabend antreten. Vervollständigen Sie Tabelle 3 und legen Sie einen geeigneten Undo-Puffer (Zeitstempel, Attribut, Undo-Image) an. Wann muss Tb committen, damit Td bereits die Änderung von Tb liest? Was lesen Ta und Tb ?
3. Brinkmann und House wollen zeitgleich den Feierabend antreten. House startet bei $T8$, Brinkmann bei $T9$. Wer darf gehen? Wie sorgt *Precision Locking* dafür, dass nur ein Arzt das Krankenhaus verlässt? Vervollständigen Sie die Einträge.



Hauptspeicher-Datenbanken

Hauptspeicher-Datenbanken

ART-Tree (Adaptiver Radix-Baum)



Hauptspeicher-Datenbanken

ART-Knotentypen

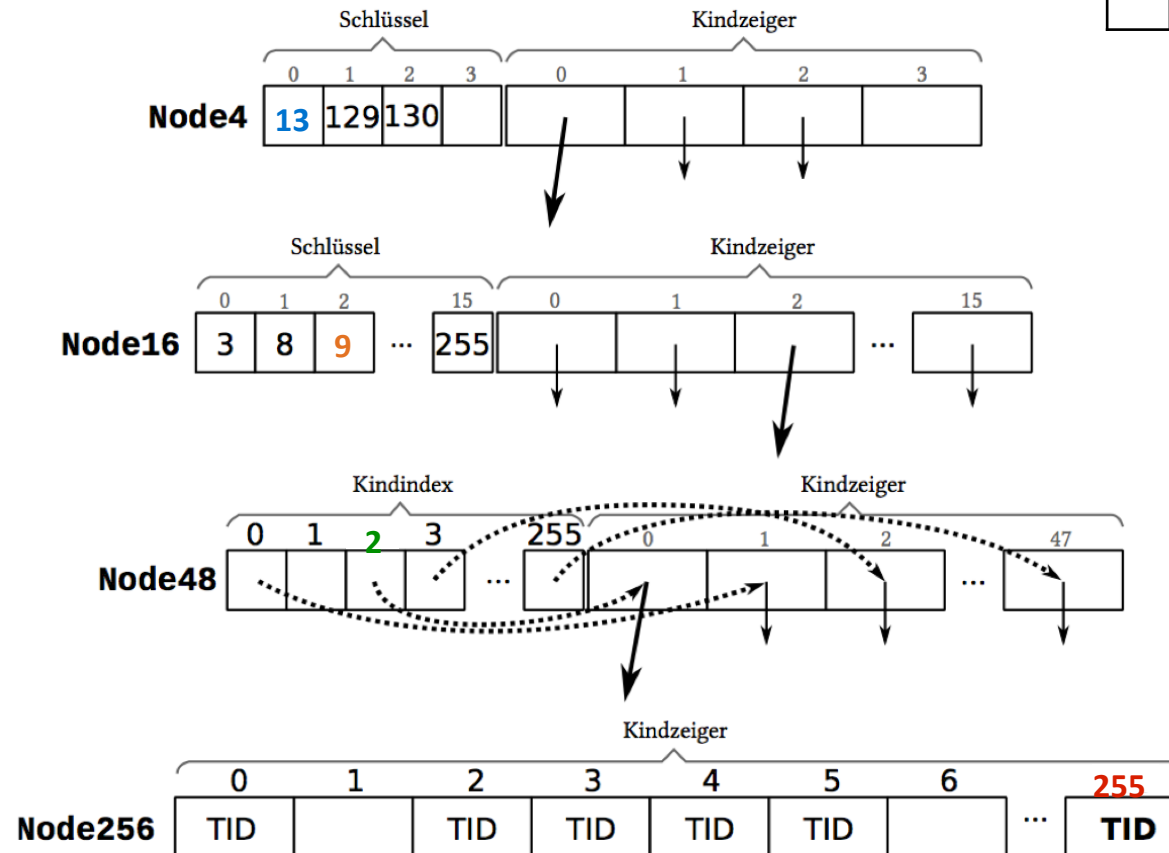
Integer Schlüssel
+218694399

Bit Repräsentation (32 bit, ohne Vorzeichen)

00001101 00001001 00000010 11111111

Byte-Repräsentation

13	9	2	255
			5

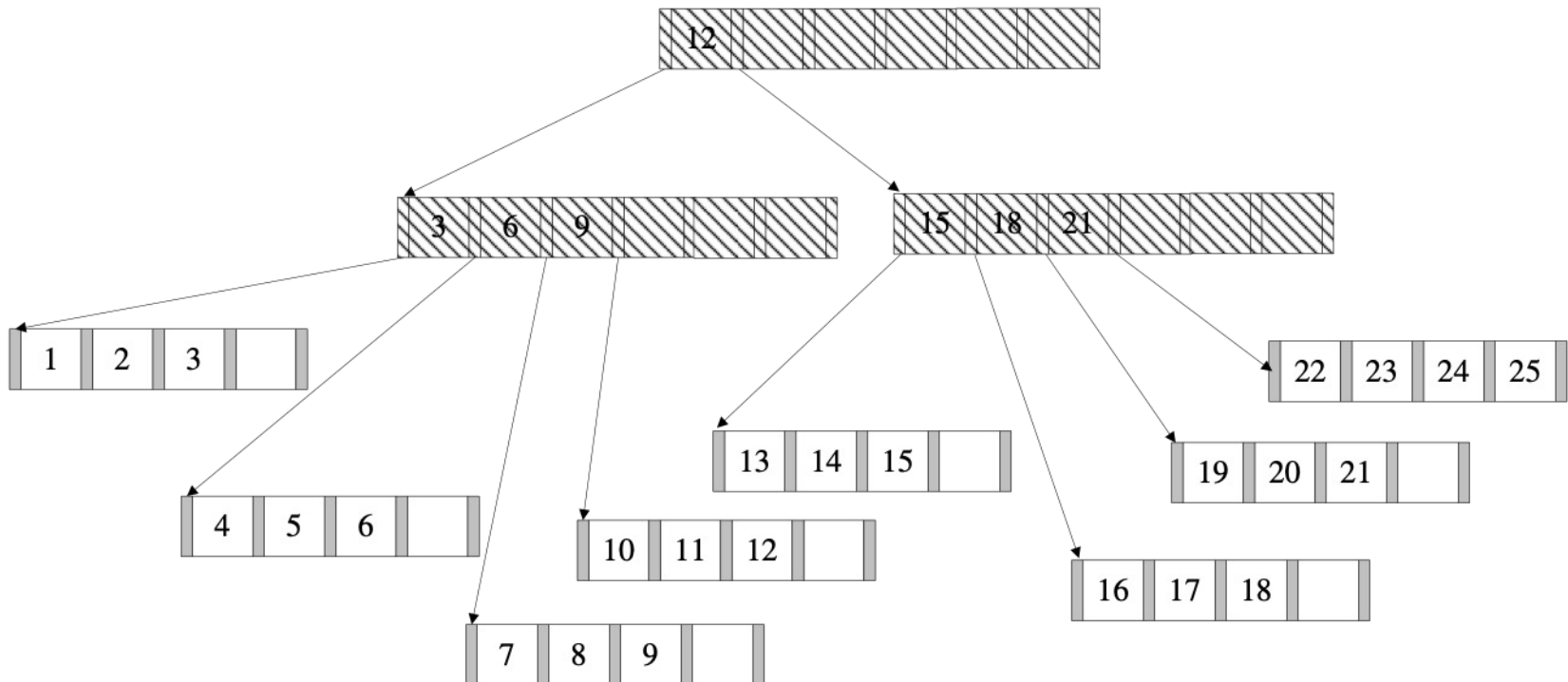


Aufgabe 3

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 3

B⁺ Baum Beispiel



Aufgabe 4

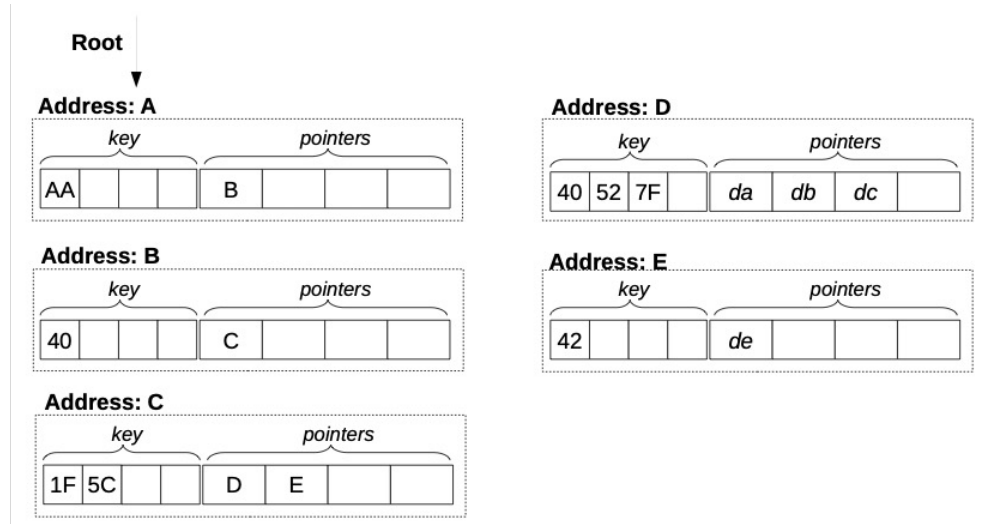


Abbildung 1: Knoten des ART (jeweils Node4)

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.



XML

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>
  </Student>
</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>
    <Senatsvorsitzender>Alex</Senat>
  </UniLeitung>
  <Studenten />
  <Student>
    <Name Peter Name>
    <MatrNr>03670815</MatrNr>
    <Vorlesungen>V1<V2<V3</Vorlesungen>
  </Studenten>
</Uni>

```

Öffnender und schließender Tag ungleich

Signalisiert leeren Tag, hat aber Inhalt

Schließender Tag fehlt

Einfach falsch

< 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

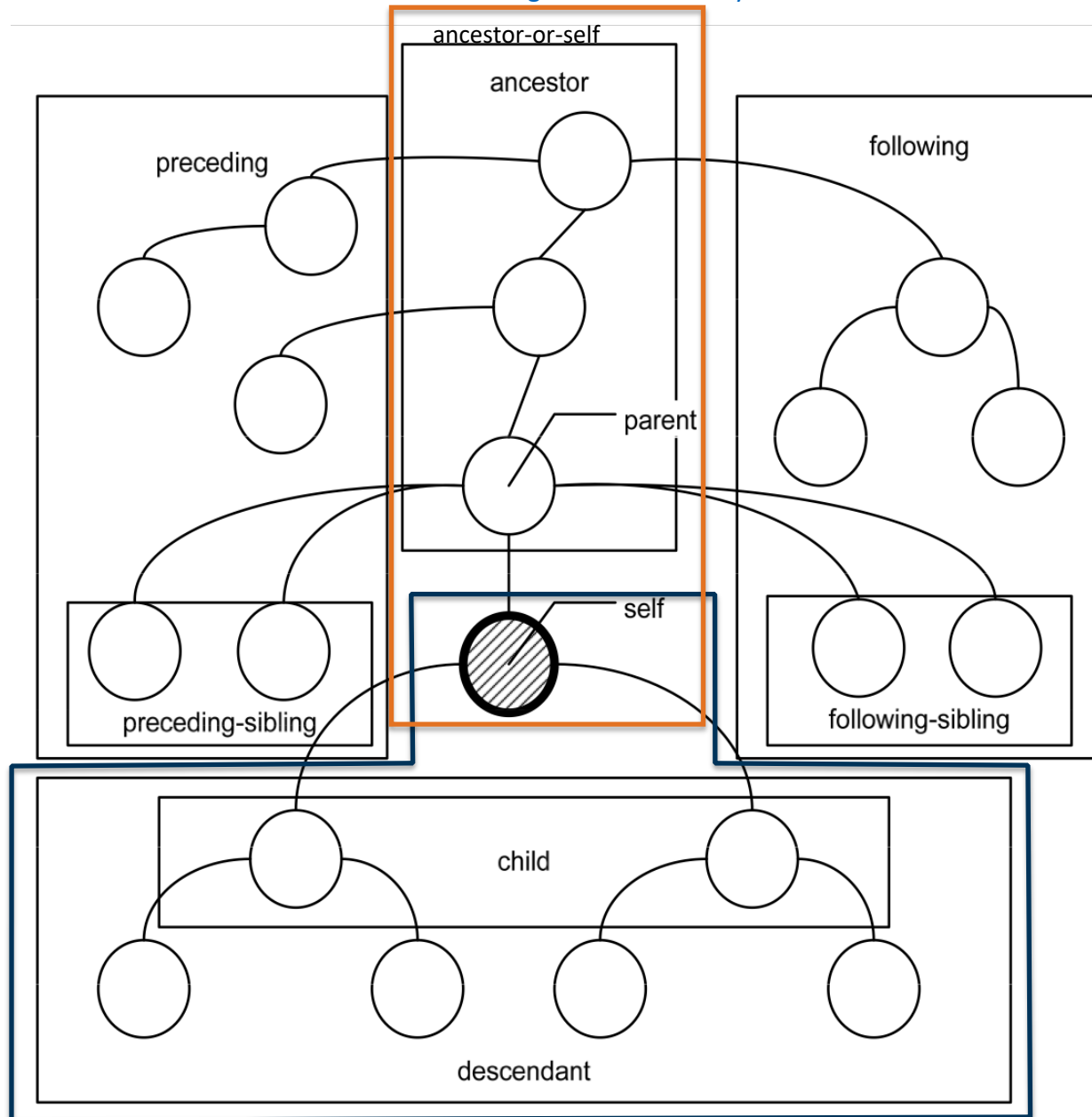
Ausgabe ist Text oder XML-Knoten

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>
  ...

```

```

</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>
  ...
</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>
  ...
</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>
```

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

Ausgabe:

```
<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>
```

```
<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?