



Einsatz und Realisierung von Datenbanksystemen

ERDB Übungsleitung
i3erdb@in.tum.de

Folien erstellt von Maximilian Bandle & Alexander Beischl



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.

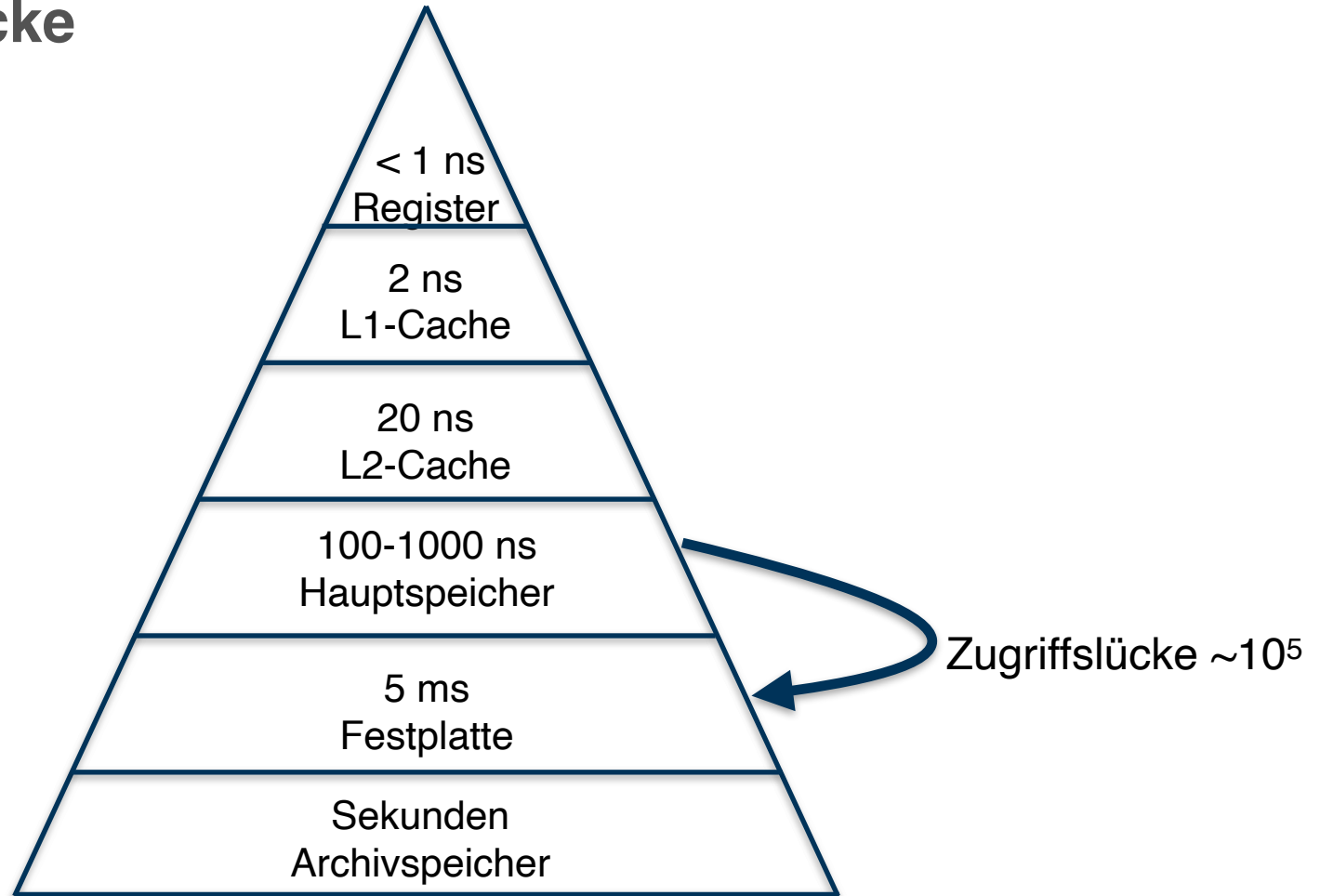
Falls Ihr einen Fehler oder eine Unstimmigkeit findet, schreibt an i3erdb@in.tum.de mit Angabe der Foliennummer.



Hauptspeicher-Datenbanken

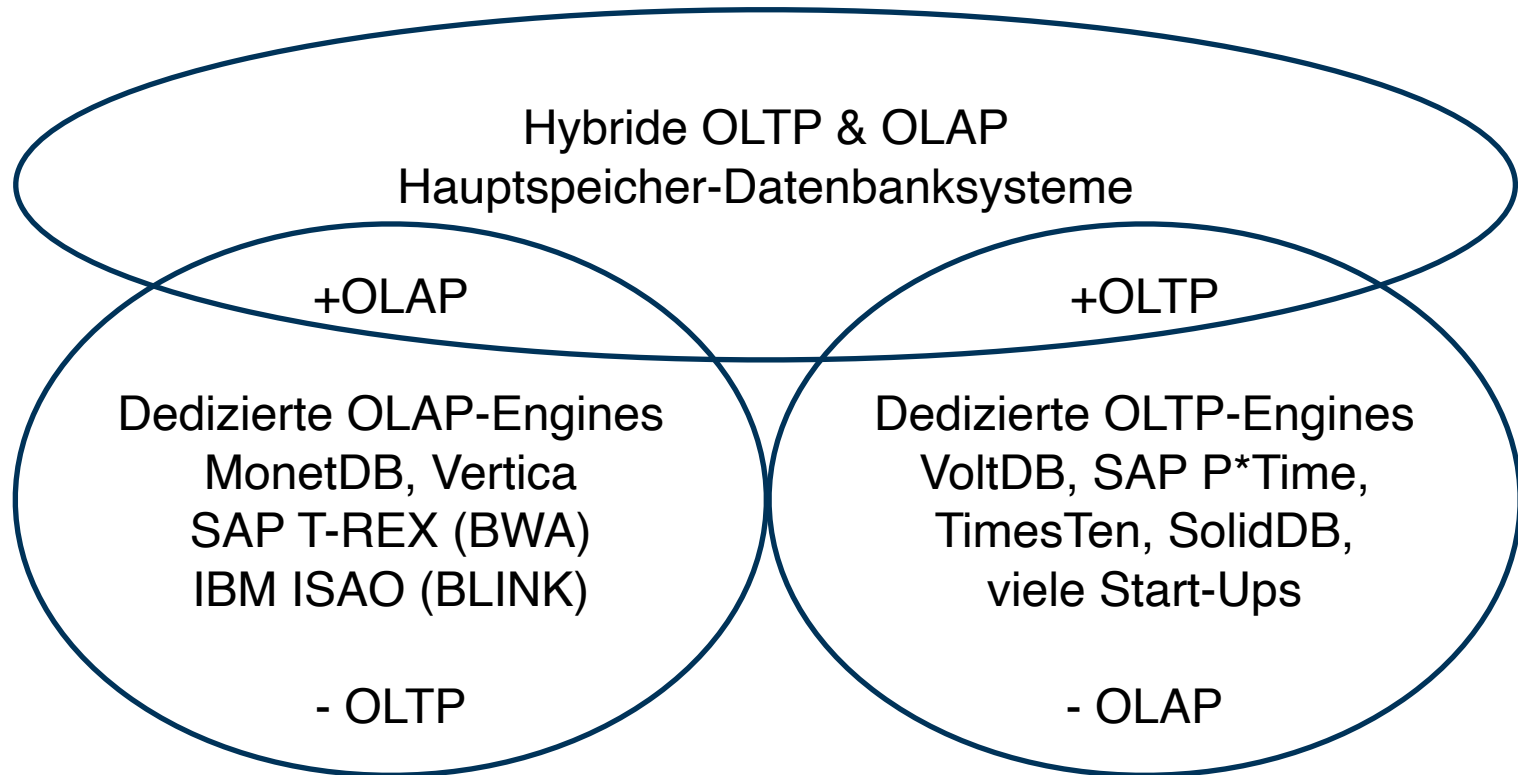
Hauptspeicher-Datenbanken

Zugriffslücke



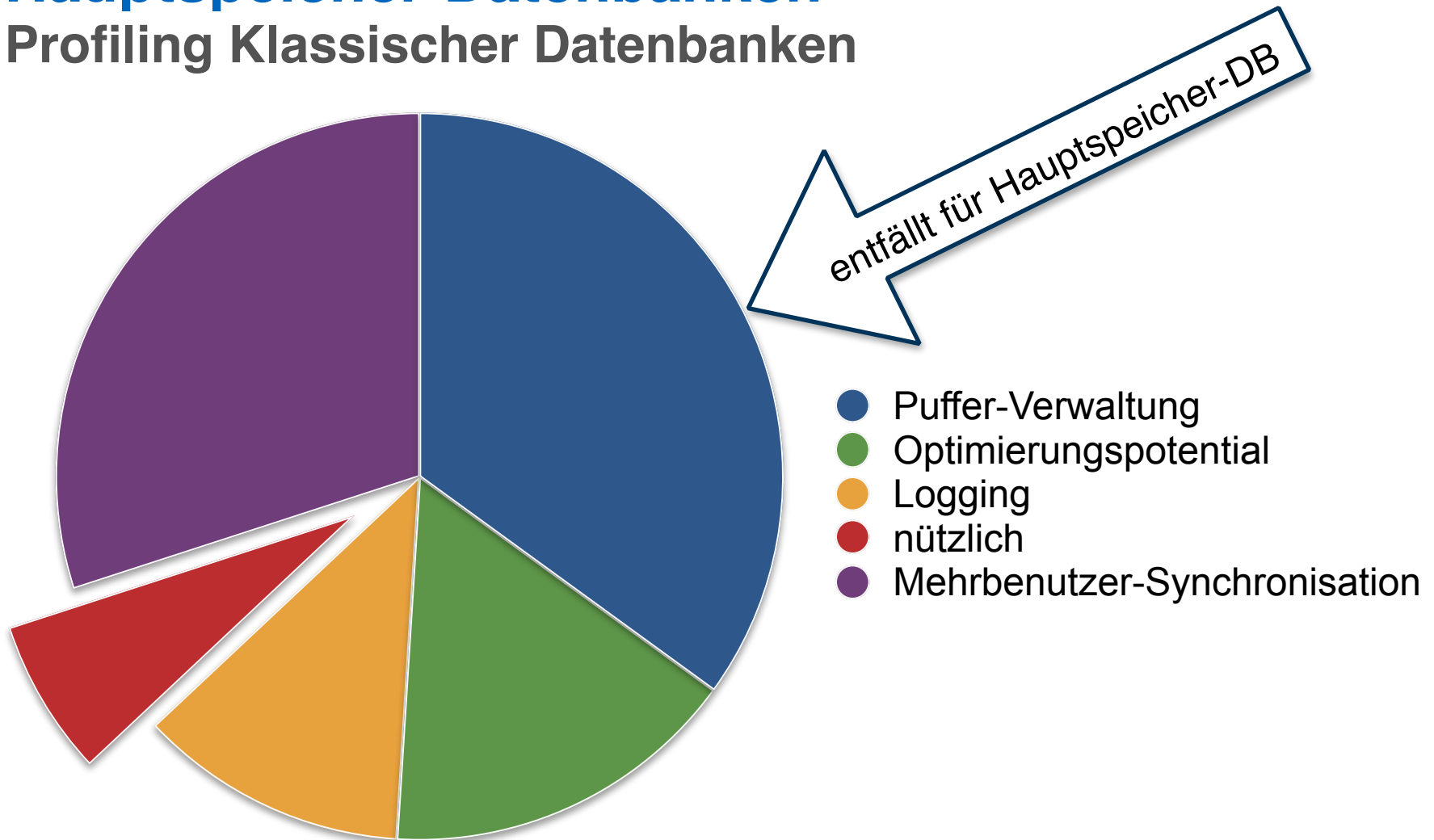
Hauptspeicher-Datenbanken

Einsatz



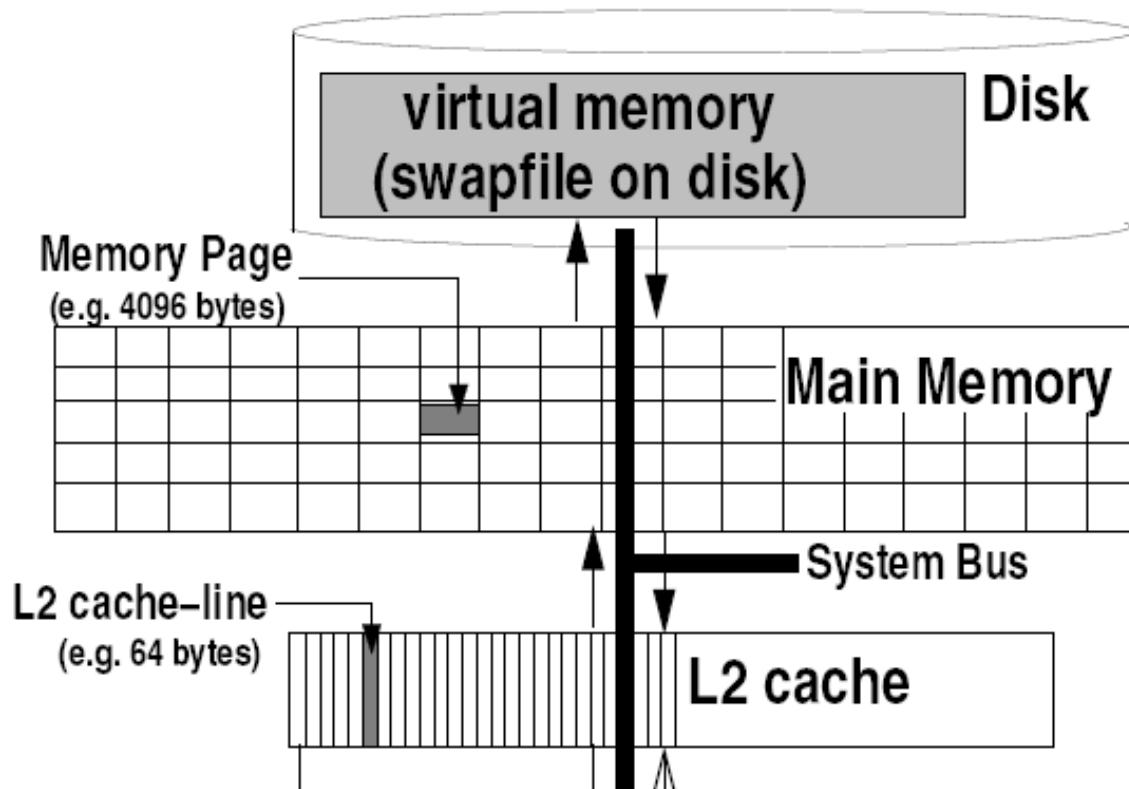
Hauptspeicher-Datenbanken

Profiling Klassischer Datenbanken



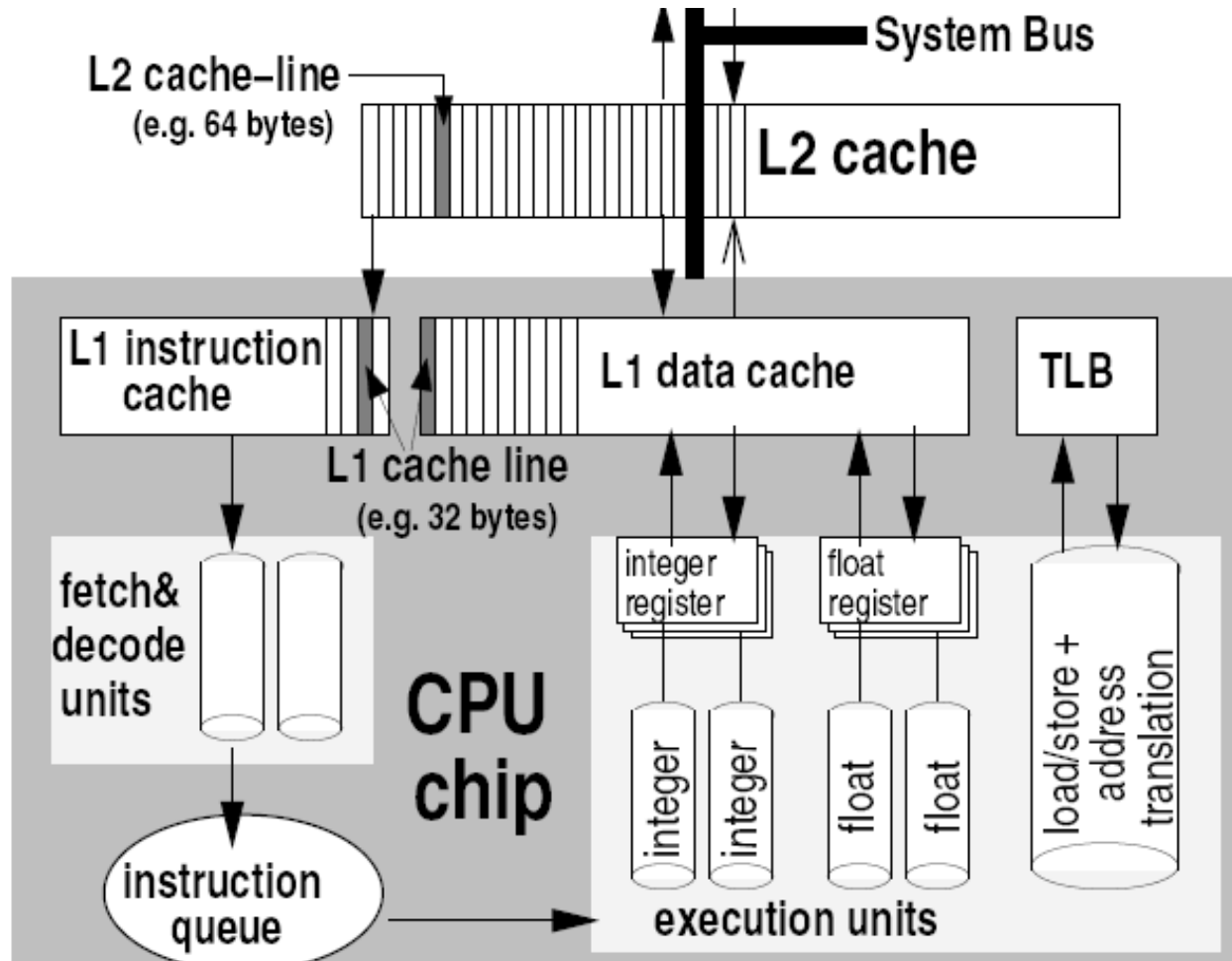
Hauptspeicher-Datenbanken

Speicherhierarchie



Hauptspeicher-Datenbanken

Speicherhierarchie

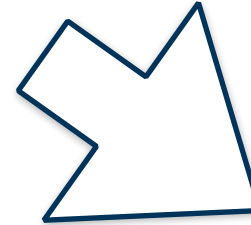
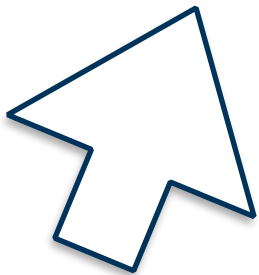


Hauptspeicher-Datenbanken

Row- vs Column-Store

Row Store

Name	MatrNr	Semester	Fach	Nebenfach
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	362101	10	Info	Mathe

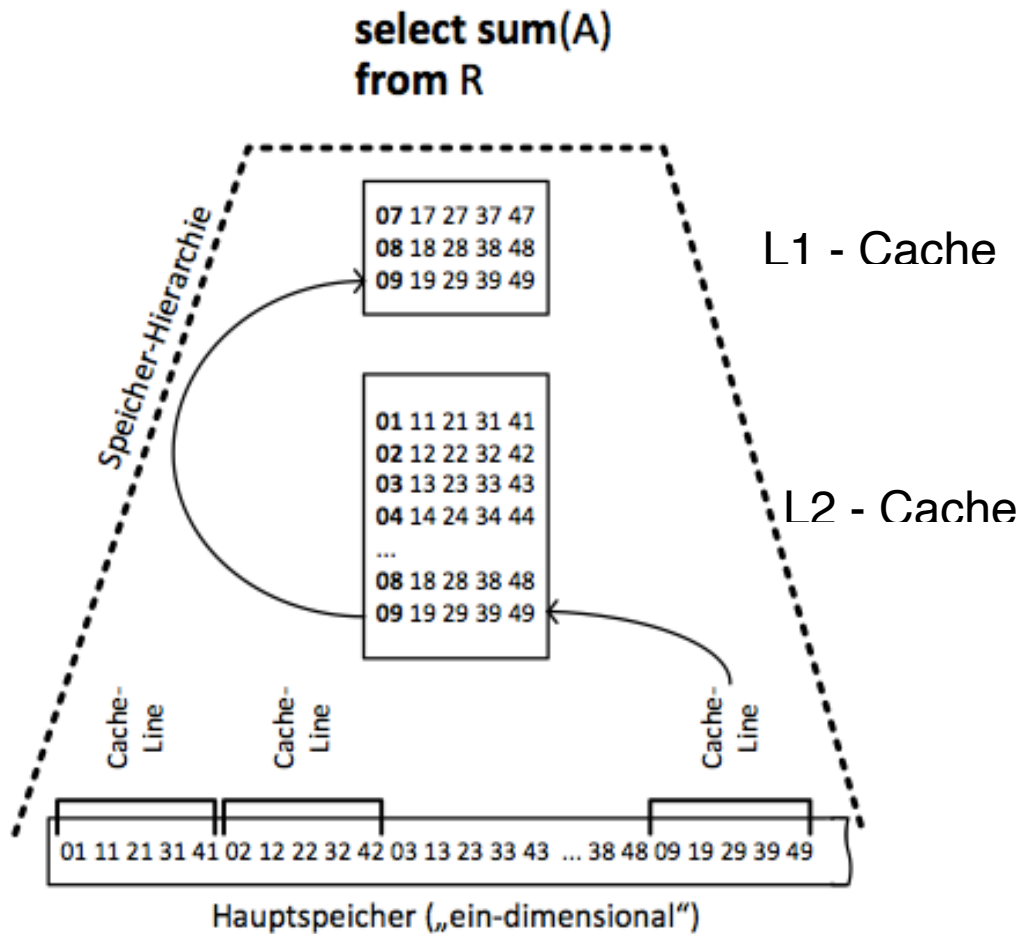


Column Store

Name	MatrNr	Semester	Fach	Nebenfach
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	362101	10	Info	Mathe

Hauptspeicher-Datenbanken

Row-Store

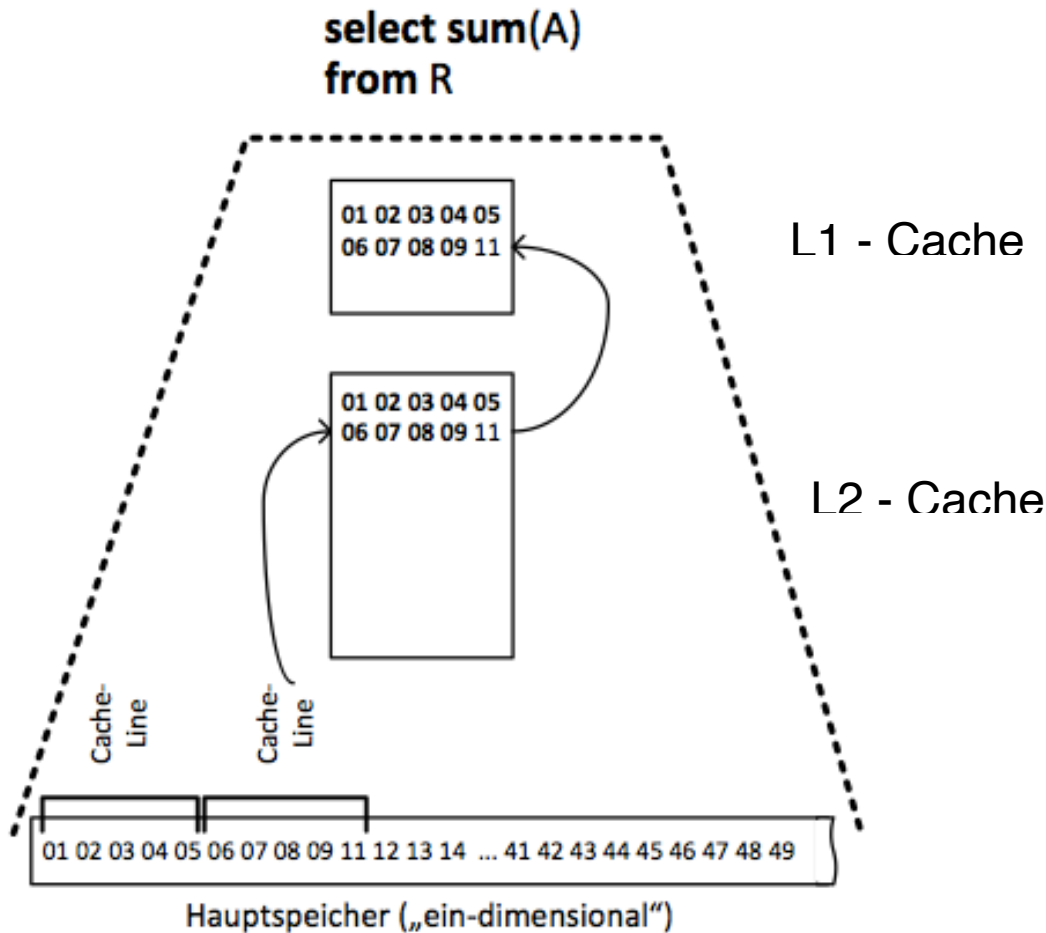


Speicherstruktur

A	B	C	D	E
01	11	21	31	41
02	12	22	32	42
03	13	23	33	43
04	14	24	34	44
05	15	25	35	45
06	16	26	36	46
07	17	27	37	47
08	18	28	38	48
09	19	29	39	49

Hauptspeicher-Datenbanken

Column-Store



Speicherstruktur

A	B	C	D	E
01	11	21	31	41
02	12	22	32	42
03	13	23	33	43
04	14	24	34	44
05	15	25	35	45
06	16	26	36	46
07	17	27	37	47
08	18	28	38	48
09	19	29	39	49

Hauptspeicher-Datenbanken

Row vs Column Store

Die Anzahl der Tupel in der Relation Studenten ist nicht bekannt, wir verwenden ISI als Abschätzung.
Für die MatrNr existiert ein Index.

Row Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...				

Column Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...	...	...	...	...

Hauptspeicher-Datenbanken

Row vs Column Store

```
select *
from Studenten;
```

Row Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...				

Die Anzahl der Tupel in der Relation Studenten ist nicht bekannt, wir verwenden ISI als Abschätzung.

Für die MatrNr existiert ein Index. 1 B = 1 Byte (8Bit)

Column Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...	...	...	...	...

Hauptspeicher-Datenbanken

Row vs Column Store

```
select *
from Studenten;
```

Row Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...				

Die Anzahl der Tupel in der Relation Studenten ist nicht bekannt, wir verwenden ISI als Abschätzung.

Für die MatrNr existiert ein Index. 1 B = 1 Byte (8Bit)

RowStore:

1 Tupel: 32B + 3B + 1B + 4B + 16B = 56B

$$\begin{aligned} \#Cachelines &= \lceil ISI * (56\text{Byte}/64\text{Byte}) \rceil \\ &= \lceil ISI * (7/8) \rceil \end{aligned}$$

Hauptspeicher-Datenbanken

Row vs Column Store

```
select *
from Studenten;
```

ColumnStore:

$$\begin{aligned}
 \#Cachelines &= \lceil ISI \cdot (32B/64B) \rceil + \lceil ISI \cdot (3B/64B) \rceil + \\
 &\quad \lceil ISI \cdot (1B/64B) \rceil + \lceil ISI \cdot (4B/64B) \rceil + \lceil ISI \cdot (16B/64B) \rceil \\
 &= \lceil ISI \cdot (32B + 3B + 1B + 4B + 16B) / 64B \rceil \\
 &= \lceil ISI \cdot 56B / 64B \rceil \\
 &= \lceil ISI \cdot 7/8 \rceil
 \end{aligned}$$

Die Anzahl der Tupel in der Relation Studenten ist nicht bekannt, wir verwenden ISI als Abschätzung.

Für die MatrNr existiert ein Index. 1 B = 1 Byte (8Bit)

Column Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...	...	...	...	...

Hauptspeicher-Datenbanken

Row vs Column Store

```
select Name, MatrNr
from Studenten
where Semester = 10;
```

Row Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...				

Die Anzahl der Tupel in der Relation Studenten ist nicht bekannt, wir verwenden ISI als Abschätzung.

Für die MatrNr existiert ein Index. 1 B = 1 Byte (8Bit)

Column Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...	...	...	...	...

Hauptspeicher-Datenbanken

Row vs Column Store

```
select Name, MatrNr
from Studenten
where Semester = 10;
```

Row Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...				

Die Anzahl der Tupel in der Relation Studenten ist nicht bekannt, wir verwenden ISI als Abschätzung.

Für die MatrNr existiert ein Index. 1 B = 1 Byte (8Bit)

RowStore:

$$\begin{aligned}
 \#Cachelines &= \lceil ISI * (56\text{Byte}/64\text{Byte}) \rceil \\
 &= \lceil ISI * (7/8) \rceil \\
 &= \lceil ISI * 0,875 \rceil
 \end{aligned}$$

Hauptspeicher-Datenbanken

Row vs Column Store

```
select Name, MatrNr
from Studenten
where Semester = 10;
```

ColumnStore:

$$\begin{aligned}
 \#Cachelines &= \lceil ISI * 1B/64B \rceil + \lceil ISI * 32B/64B * \\
 &\quad 1/10 \rceil + \lceil ISI * 3B/64B * 1/10 \rceil \\
 &= \lceil ISI * (1B/64B + 32B/640B + 3B/640B) \rceil \\
 &= \lceil ISI * (10B + 32B + 3B)/640B \rceil \\
 &= \lceil ISI * 45/640 \rceil \\
 &= \lceil ISI * 0,070 \rceil
 \end{aligned}$$

Die Anzahl der Tupel in der Relation Studenten ist nicht bekannt, wir verwenden ISI als Abschätzung.

Für die MatrNr existiert ein Index. 1 B = 1 Byte (8Bit)

Schätzung der Selektivität von 1/10 ist unrealistisch, insbesondere die Folge das nur 1/10 der CLs gelesen werden. Erfüllt nur den Zweck eines Beispiels.

Column Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...	...	...	...	...

Hauptspeicher-Datenbanken

Row vs Column Store

```
select Name, MatrNr
from Studenten
where MatrNr = %;
```

Row Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...				

Die Anzahl der Tupel in der Relation Studenten ist nicht bekannt, wir verwenden ISI als Abschätzung.

Für die MatrNr existiert ein Index. 1 B = 1 Byte (8Bit)

Column Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...	...	...	...	...

Hauptspeicher-Datenbanken

Row vs Column Store

```
select Name, MatrNr
from Studenten
where MatrNr = %;
```

Row Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...				

Die Anzahl der Tupel in der Relation Studenten ist nicht bekannt, wir verwenden ISI als Abschätzung.

Für die MatrNr existiert ein Index. 1 B = 1 Byte (8Bit)

RowStore:

$$\#Cachelines = \lceil 56B/64B \rceil = 1$$

Hier wird der Index von MatrNr genutzt. Deshalb muss nur das Tupel mit der gesuchten MatrNr geladen werden. Dieser umfasst 1 Cacheline.

Hauptspeicher-Datenbanken

Row vs Column Store

```
select Name, MatrNr
from Studenten
where MatrNr = %;
```

ColumnStore:

$$\#Cachelines = \lceil \frac{32B}{64B} \rceil + \lceil \frac{3B}{64B} \rceil = 2$$

Hier wird ebenfalls wieder der Index von MatrNr genutzt, sodass nur der Namen und die MatrNr des Tupels mit der gesuchten MatrNr aus den jeweiligen Tabellengeladen wird.

Die Anzahl der Tupel in der Relation Studenten ist nicht bekannt, wir verwenden ISI als Abschätzung.

Für die MatrNr existiert ein Index. 1 B = 1 Byte (8Bit)

Column Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...	...	...	...	...

Hauptspeicher-Datenbanken

Row vs Column Store

Insert into Studenten VALUES(...);

Row Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...				

Die Anzahl der Tupel in der Relation Studenten ist nicht bekannt, wir verwenden ISI als Abschätzung.

Für die MatrNr existiert ein Index. 1 B = 1 Byte (8Bit)

Column Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...	...	...	...	...

Hauptspeicher-Datenbanken

Row vs Column Store

Insert into Studenten VALUES(...);

Row Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...				

RowStore:

$$\#Cachelines = \lceil 56B/64B \rceil = 1$$

Die Anzahl der Tupel in der Relation Studenten ist nicht bekannt, wir verwenden ISI als Abschätzung.

Für die MatrNr existiert ein Index. 1 B = 1 Byte (8Bit)

Hauptspeicher-Datenbanken

Row vs Column Store

Insert into Studenten VALUES(...);

ColumnStore:

$$\#Cachelines = \lceil \frac{32B}{64B} \rceil + \lceil \frac{3B}{64B} \rceil + \lceil \frac{1B}{64B} \rceil + \lceil \frac{4B}{64B} \rceil + \lceil \frac{16B}{64B} \rceil = 5$$

Da jedes Attribut muss einzeln in die jeweilige Tabelle eingefügt werden.

Die Anzahl der Tupel in der Relation Studenten ist nicht bekannt, wir verwenden ISI als Abschätzung.

Für die MatrNr existiert ein Index. 1 B = 1 Byte (8Bit)

Column Store

Name (32Byte)	MatrNr (3Byte)	Semester (1 Byte)	Fach (4Byte)	Nebenfach (16 Byte)
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	262101	10	Info	Mathe
...	...	...	...	...

Aufgabe 1

Gegeben eine Tabelle *Produkte* mit folgendem Schema und 10000 Einträgen:

Id (8 Byte) | Name (32 Byte) | Preis (8 Byte) | Anzahl (8 Byte)

Wieviele Daten werden für folgende Queries in die CPU-Caches geladen? Unterscheiden sie jeweils zwischen Row und Column Store.

1. *select * from Produkte*
2. *select Anzahl from Produkte*

Aufgabe 2

Sie sollen für die Alexander-Maximilians-Universität (AMU) ein Hauptspeicherdatenbanksystem optimieren. In dem System sind die Daten aller Studenten gespeichert. Schätzen Sie für jede der untenstehenden Anfragen einzeln, ob ein Row- oder Column-Store besser geeignet ist.

Relationen

Studenten: MatrNr (8 Byte), Name (48 Byte), Studiengang (4 Byte), Semester (4 Byte)
MatrNr ist der Primärschlüssel der indiziert ist.

Anfragen:

1. `select * from Studenten;`
2. `select Semester, count(*) from Studenten group by Semester;`
3. `select Name, Studiengang, Semester from Studenten where MatrNr = 42;`
4. `select Studiengang from Studenten where MatrNr = 42;`
5. `select * from Studenten where Semester < 5;`
6. `select * from Studenten where Semester = 25;`
7. `insert into studenten values(4242, Max Meyer, Info, 7);`

Aufgabe 3

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?



XML-Anfragesprachen (XQuery)

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 4

Formulieren Sie die zuvor in SQL bearbeiteten Anfragen zur Universitätsdatenbank in XQuery. Erstellen Sie insbesondere XQuery-Anfragen, um folgende Fragestellungen zu beantworten ²:

- a) Suchen Sie die Professoren, die Vorlesungen halten.
- b) Finden Sie die Studenten, die alle Vorlesungen gehört haben.
- c) Finden Sie die Studenten mit der größten Semesterzahl unter Verwendung von Aggregatfunktionen.
- d) Berechnen Sie die Gesamtzahl der Semesterwochenstunden, die die einzelnen Professoren erbringen. Dabei sollen auch die Professoren berücksichtigt werden, die keine Vorlesungen halten.
- e) Finden Sie die Studenten, die alle vierstündigen Vorlesungen gehört haben.
- f) Finden Sie die Namen der Studenten, die in keiner Prüfung eine bessere Note als 3.0 hatten.
- g) Berechnen Sie den Umfang des Prüfungsstoffes jedes Studenten. Es sollen der Name des Studenten und die Summe der Semesterwochenstunden der Prüfungsvorlesungen ausgegeben werden.
- h) Finden Sie Studenten, deren Namen den eines Professors enthalten.
- i) Ermitteln Sie den Bekanntheitsgrad der Professoren unter den Studenten, wobei wir annehmen, dass Studenten die Professoren nur durch Vorlesungen oder Prüfungen kennen lernen.

Aufgabe 3

```

<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>

```

a) Suchen Sie die Professoren, die Vorlesungen halten.

Aufgabe 3

```
<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>
```

b) Finden Sie die Studenten, die alle Vorlesungen gehört haben.

Aufgabe 3

```
<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>
```

c) Finden Sie die Studenten mit der größten Semesterzahl unter Verwendung von Aggregatfunktionen.

Aufgabe 3

```
<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>
      </ProfessorIn>
    </Fakultaet>
  </Fakultaeten>
```

```
</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>
```

- d) Berechnen Sie die Gesamtzahl der Semesterwochenstunden, die die einzelnen Professoren erbringen. Dabei sollen auch die Professoren berücksichtigt werden, die keine Vorlesungen halten.

Aufgabe 3

```
<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>
```

e) Finden Sie die Studenten, die alle vierstündigen Vorlesungen gehört haben.

Aufgabe 3

```
<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>
```

f) Finden Sie die Namen der Studenten, die in keiner Prüfung eine bessere Note als 3.0 hatten.

Aufgabe 3

```
<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>
```

g) Berechnen Sie den Umfang des Prüfungsstoffes jedes Studenten. Es sollen der Name des Studenten und die Summe der Semesterwochenstunden der Prüfungsvorlesungen ausgegeben werden.

Aufgabe 3

```
<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>
```

h) Finden Sie Studenten, deren Namen den eines Professors enthalten.

Aufgabe 3

```

<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>

```

- i) Ermitteln Sie den Bekanntheitsgrad der Professoren unter den Studenten, wobei wir annehmen, dass Studenten die Professoren nur durch Vorlesungen oder Prüfungen kennen lernen.

Aufgabe 5

Schreiben Sie eine Anfrage, die folgendes zurück gibt:

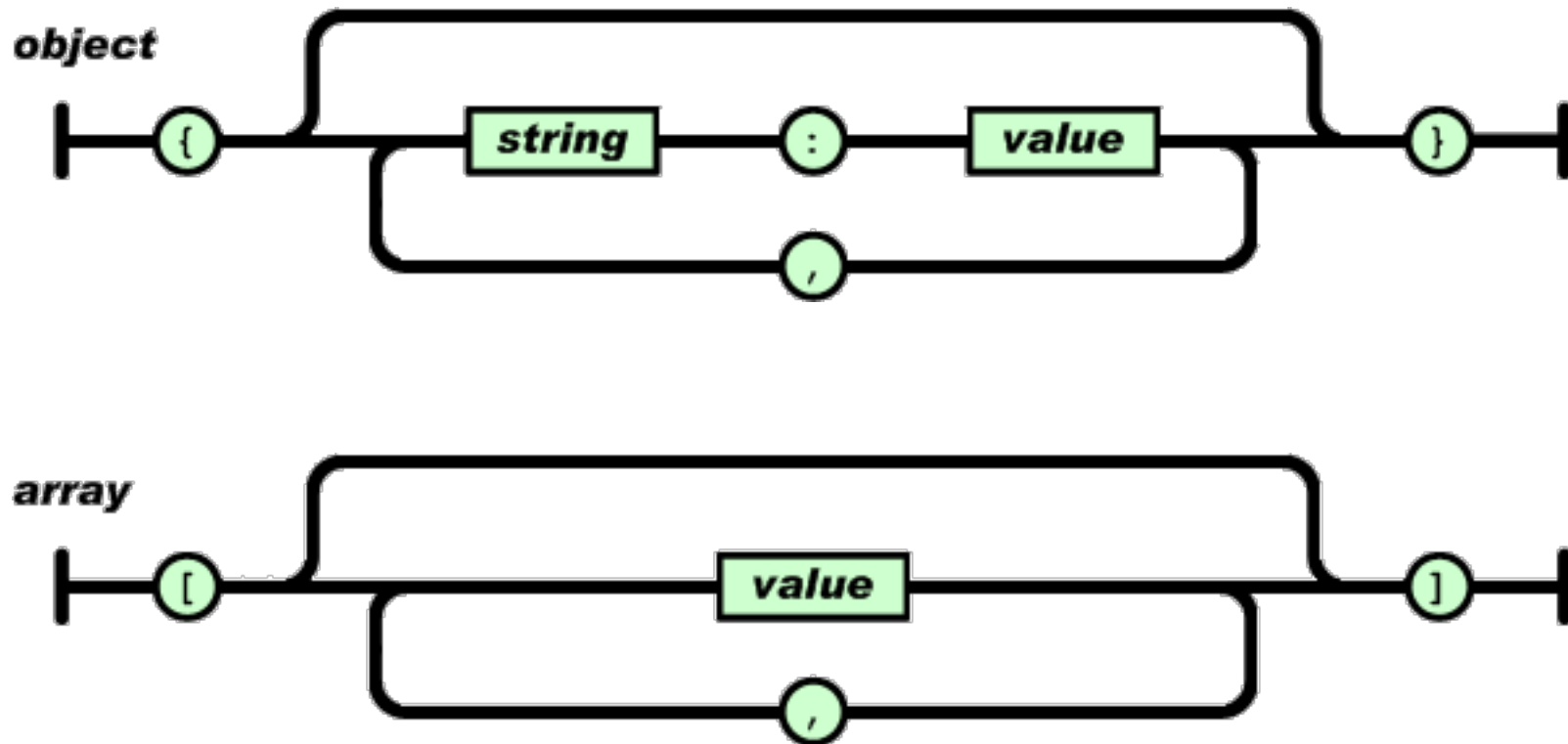
```
<Universitaet>
  <Fakultaet Name="Philosophie" AnzahlAssistenten="3">
    <Professor Name="Sokrates" AnzahlAssistenten="2"/>
    <Professor Name="Russel" AnzahlAssistenten="1"/>
  </Fakultaet>
  <Fakultaet Name="Physik" AnzahlAssistenten="2">
    <Professor Name="Kopernikus" AnzahlAssistenten="2"/>
  </Fakultaet>
  <Fakultaet Name="Theologie" AnzahlAssistenten="1">
    <Professor Name="Augustinus" AnzahlAssistenten="1"/>
  </Fakultaet>
</Universitaet>
```



JSON

- JSON baut auf zwei Strukturen auf:
 - Objekt
 - Array

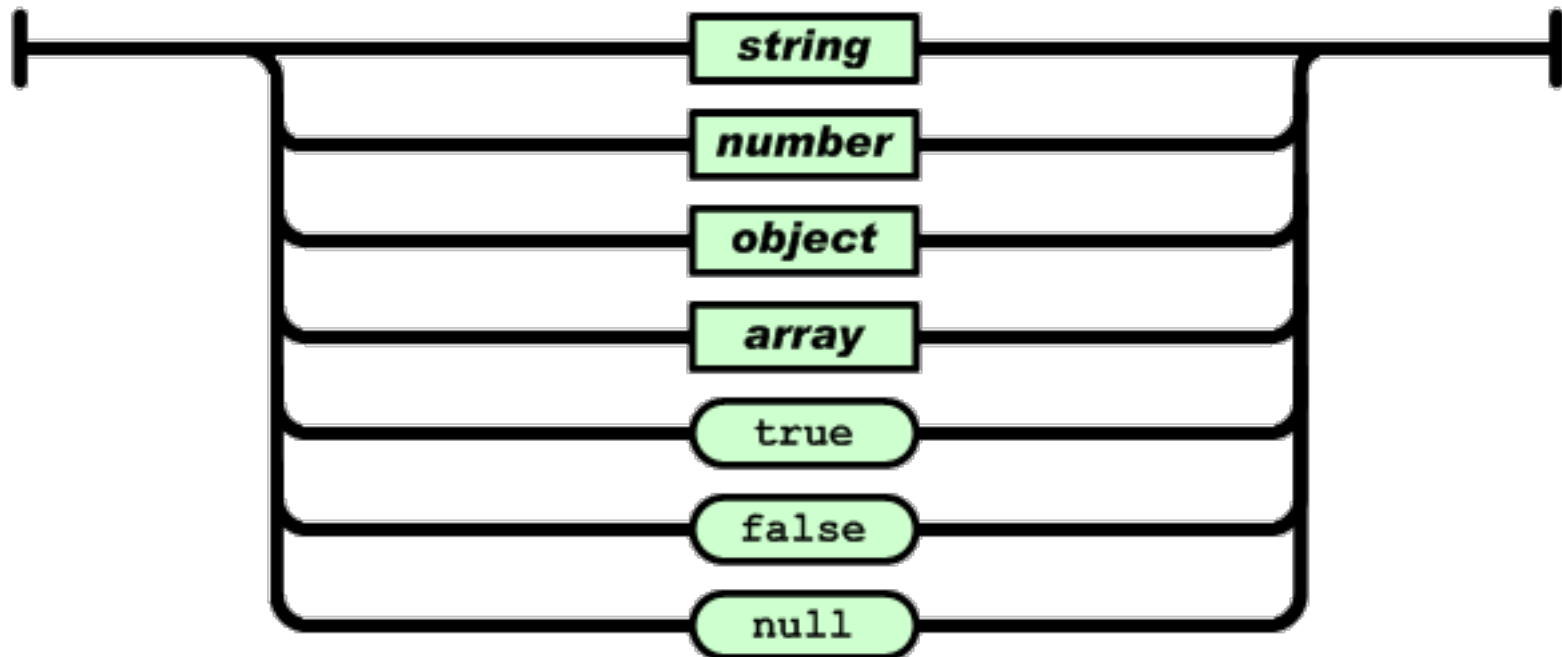
JSON



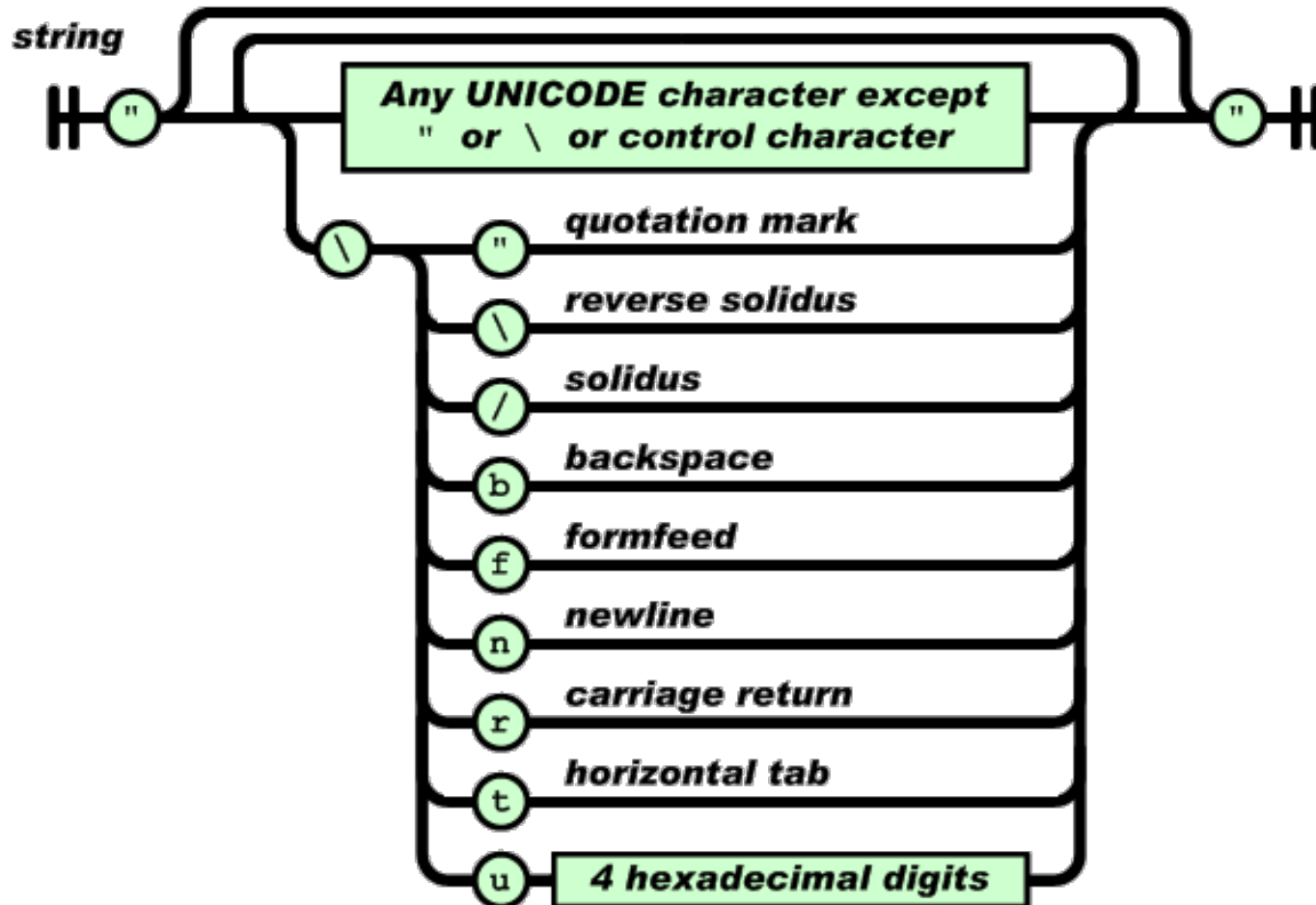


JSON

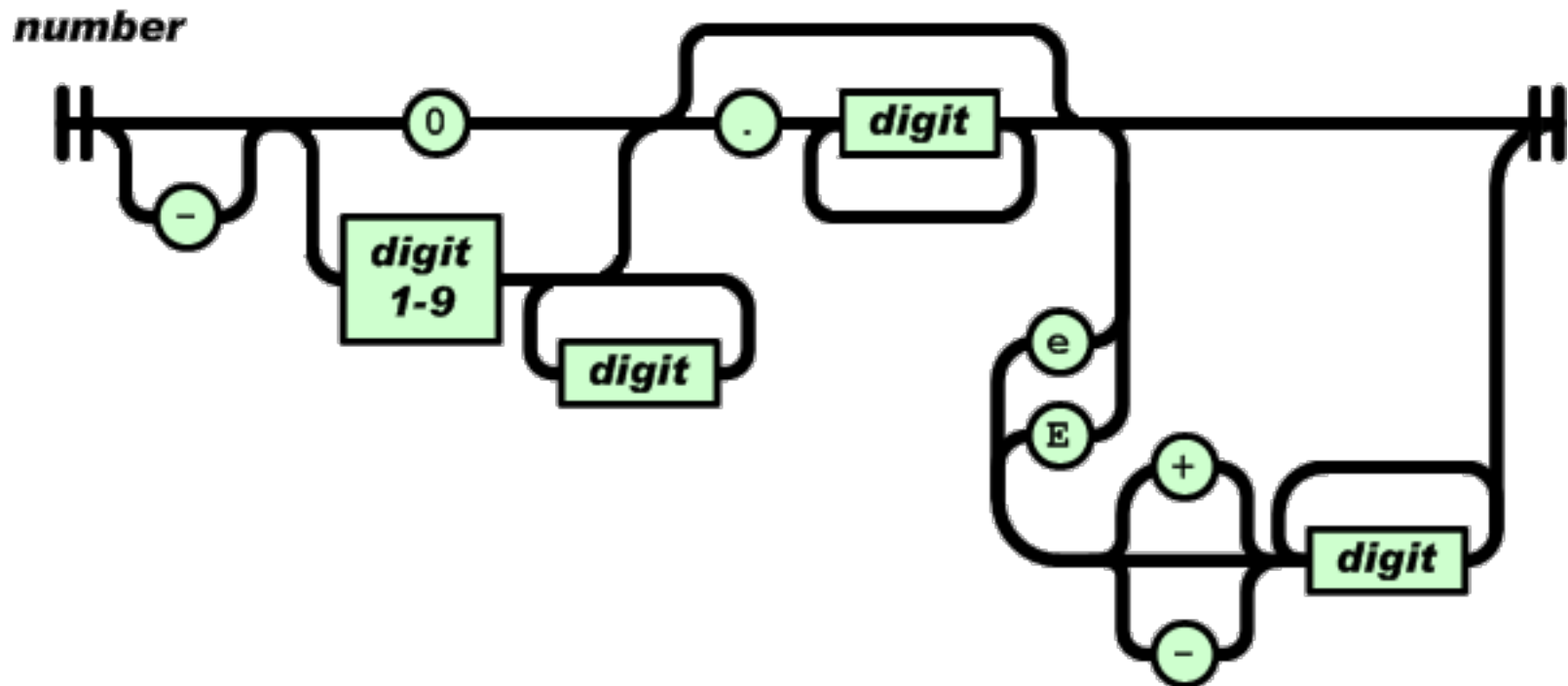
value



JSON



JSON



JSONpath: Standardisierte Anfragesprache

Beispiele:

`$..Universität.UniLeitung.Rektor`

`$['Universität']['UniLeitung']['Rektor']`

`$..Professoren..Vorlesungen..Titel`

`$..Professoren[?(@.Name=='Kopernikus')..Vorlesungen..Titel`

`$..Professoren[?(@.Name=='Kopernikus')..Vorlesungen[0].Titel`

`$..Professoren[?(@.Name=='Kopernikus')..Vorlesungen[0:2].Titel`

`$..Universität.Fakultäten[*].Professoren
[?(@.Name=='Kopernikus')..Vorlesungen[:1].Titel`

`$['Universität']['Fakultäten']['*']['Professoren']
[?(@['Name']=='Kopernikus')..['Vorlesungen'][:1]['Titel']`

```
{
  "Universität": {
    "Name": "Virtuelle Universität der Grösten De",
    "UniLeitung": {
      "Rektor": "Sokrates",
      "Kanzler": "Erhard" },
    "Fakultäten": [
      { "FakName": "Theologie",
        "Professoren": [ {
          "Name": "Augustinus",
          "Rang": "C3",
          "Raum": 309,
          "Vorlesungen": [
            { "VorlNr": "V5022",
              "Titel": "Glaube und Wissen",
              "SWS": 2 } ] ] },
        { "FakName": "Physik",
          "Professoren": [
            { "Name": "Curie",
              "Rang": "C4",
              "Raum": 36 },
            { "Name": "Kopernikus",
              "Rang": "C3",
              "Raum": 310,
              "Vorlesungen": [
                { "VorlNr": "V5023",
                  "Titel": "Alfonsinische Tafeln",
                  "SWS": 2 },
                { "VorlNr": "V5024",
                  "Titel": "Astronomie",
                  "SWS": 2 } ] ] } ] ] } ] }
```

Aufgabe 6

Beantworten Sie folgende Anfragen mithilfe von JSONpath über das nachfolgende JSON-Dokument. Zum Auswerten der Anfragen können Sie folgende Schnittstelle verwenden: <https://jsonpath.com/>

1. Geben Sie den Namen aller Produkte aus

```
{
  "shop": {
    "name": "Tech & More",
    "categories": [
      {
        "name": "Laptops",
        "products": [
          { "id": 101, "name": "Laptop A", "price": 899, "tags": ["ultrabook", "ssd"] },
          { "id": 102, "name": "Laptop B", "price": 1299, "tags": ["gaming", "16GB"] }
        ]
      },
      {
        "name": "Phones",
        "products": [
          { "id": 201, "name": "Phone X", "price": 699, "tags": ["5G", "oled"] },
          { "id": 202, "name": "Phone Y", "price": 499, "tags": ["budget", "4G"] },
          { "id": 203, "name": "Phone Z", "price": 999, "tags": ["flagship", "5G"] }
        ]
      }
    ]
  }
}
```

Aufgabe 6

Beantworten Sie folgende Anfragen mithilfe von JSONpath über das nachfolgende JSON-Dokument. Zum Auswerten der Anfragen können Sie folgende Schnittstelle verwenden: <https://jsonpath.com/>

2. Geben Sie die ersten zwei Produkte aus der Kategorie „Phones“ aus

```
{
  "shop": {
    "name": "Tech & More",
    "categories": [
      {
        "name": "Laptops",
        "products": [
          { "id": 101, "name": "Laptop A", "price": 899, "tags": ["ultrabook", "ssd"] },
          { "id": 102, "name": "Laptop B", "price": 1299, "tags": ["gaming", "16GB"] }
        ]
      },
      {
        "name": "Phones",
        "products": [
          { "id": 201, "name": "Phone X", "price": 699, "tags": ["5G", "oled"] },
          { "id": 202, "name": "Phone Y", "price": 499, "tags": ["budget", "4G"] },
          { "id": 203, "name": "Phone Z", "price": 999, "tags": ["flagship", "5G"] }
        ]
      }
    ]
  }
}
```

Aufgabe 6

Beantworten Sie folgende Anfragen mithilfe von JSONpath über das nachfolgende JSON-Dokument. Zum Auswerten der Anfragen können Sie folgende Schnittstelle verwenden: <https://jsonpath.com/>

3. Geben Sie alle Preise aus

```
{
  "shop": {
    "name": "Tech & More",
    "categories": [
      {
        "name": "Laptops",
        "products": [
          { "id": 101, "name": "Laptop A", "price": 899, "tags": ["ultrabook", "ssd"] },
          { "id": 102, "name": "Laptop B", "price": 1299, "tags": ["gaming", "16GB"] }
        ]
      },
      {
        "name": "Phones",
        "products": [
          { "id": 201, "name": "Phone X", "price": 699, "tags": ["5G", "oled"] },
          { "id": 202, "name": "Phone Y", "price": 499, "tags": ["budget", "4G"] },
          { "id": 203, "name": "Phone Z", "price": 999, "tags": ["flagship", "5G"] }
        ]
      }
    ]
  }
}
```

Aufgabe 6

Beantworten Sie folgende Anfragen mithilfe von JSONpath über das nachfolgende JSON-Dokument. Zum Auswerten der Anfragen können Sie folgende Schnittstelle verwenden: <https://jsonpath.com/>

4. Geben Sie alle Produkt-Tags aus von Produkten mit einem Preis < 1000€

```
{
  "shop": {
    "name": "Tech & More",
    "categories": [
      {
        "name": "Laptops",
        "products": [
          { "id": 101, "name": "Laptop A", "price": 899, "tags": ["ultrabook", "ssd"] },
          { "id": 102, "name": "Laptop B", "price": 1299, "tags": ["gaming", "16GB"] }
        ]
      },
      {
        "name": "Phones",
        "products": [
          { "id": 201, "name": "Phone X", "price": 699, "tags": ["5G", "oled"] },
          { "id": 202, "name": "Phone Y", "price": 499, "tags": ["budget", "4G"] },
          { "id": 203, "name": "Phone Z", "price": 999, "tags": ["flagship", "5G"] }
        ]
      }
    ]
  }
}
```

Aufgabe 6

Beantworten Sie folgende Anfragen mithilfe von JSONpath über das nachfolgende JSON-Dokument. Zum Auswerten der Anfragen können Sie folgende Schnittstelle verwenden: <https://jsonpath.com/>

5. Geben Sie alle Produkte aus mit einem 5G Tag

```
{
  "shop": {
    "name": "Tech & More",
    "categories": [
      {
        "name": "Laptops",
        "products": [
          { "id": 101, "name": "Laptop A", "price": 899, "tags": ["ultrabook", "ssd"] },
          { "id": 102, "name": "Laptop B", "price": 1299, "tags": ["gaming", "16GB"] }
        ]
      },
      {
        "name": "Phones",
        "products": [
          { "id": 201, "name": "Phone X", "price": 699, "tags": ["5G", "oled"] },
          { "id": 202, "name": "Phone Y", "price": 499, "tags": ["budget", "4G"] },
          { "id": 203, "name": "Phone Z", "price": 999, "tags": ["flagship", "5G"] }
        ]
      }
    ]
  }
}
```



Fragen?