



Einsatz und Realisierung von Datenbanksystemen

ERDB Übungsleitung

Alice Rey, Maximilian Bandle, Michael Jungmair

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.

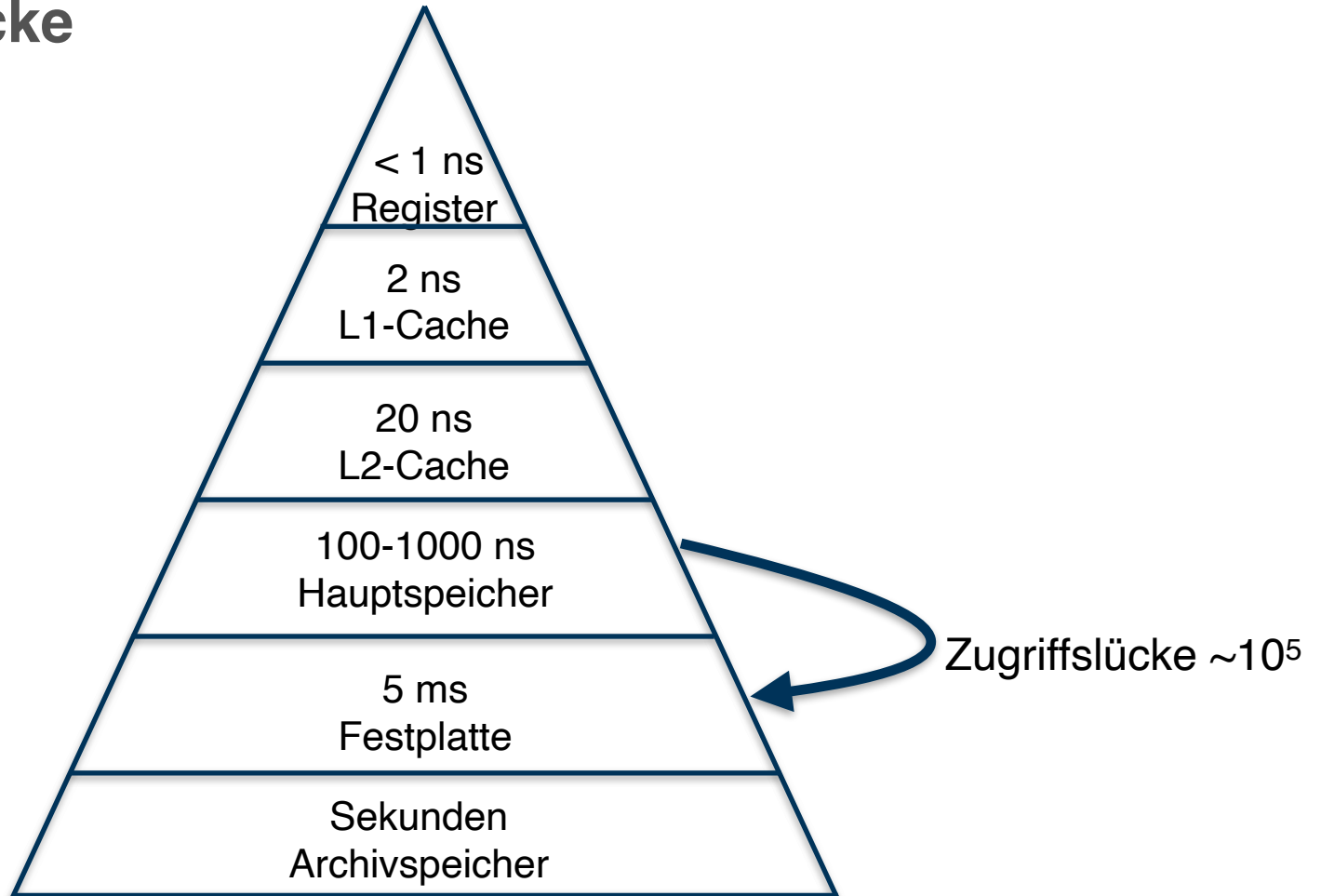
Falls Ihr einen Fehler oder eine Unstimmigkeit findet, schreibt an i3erdb@in.tum.de mit Angabe der Foliennummer.



Hauptspeicher-Datenbanken

Hauptspeicher-Datenbanken

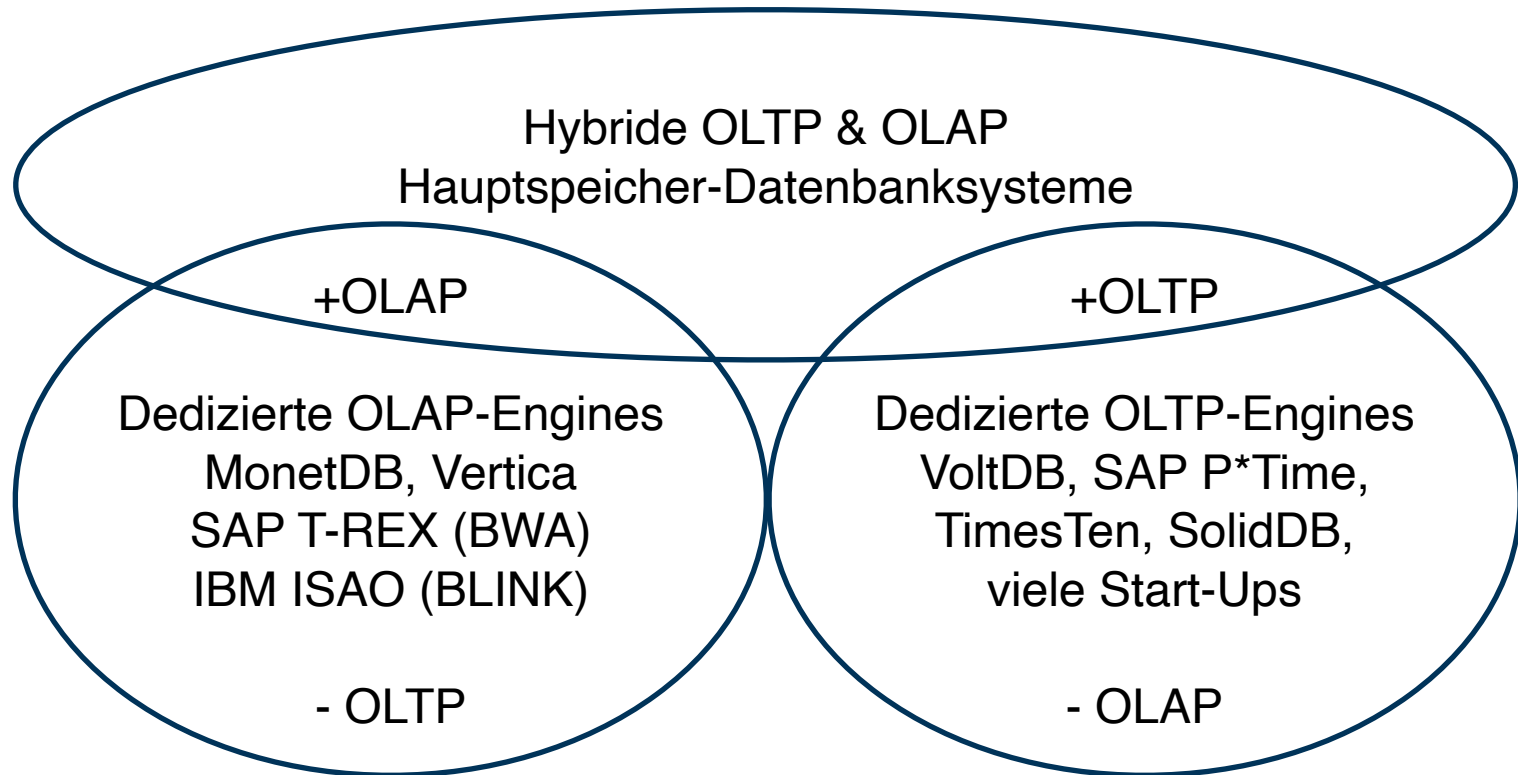
Zugriffslücke





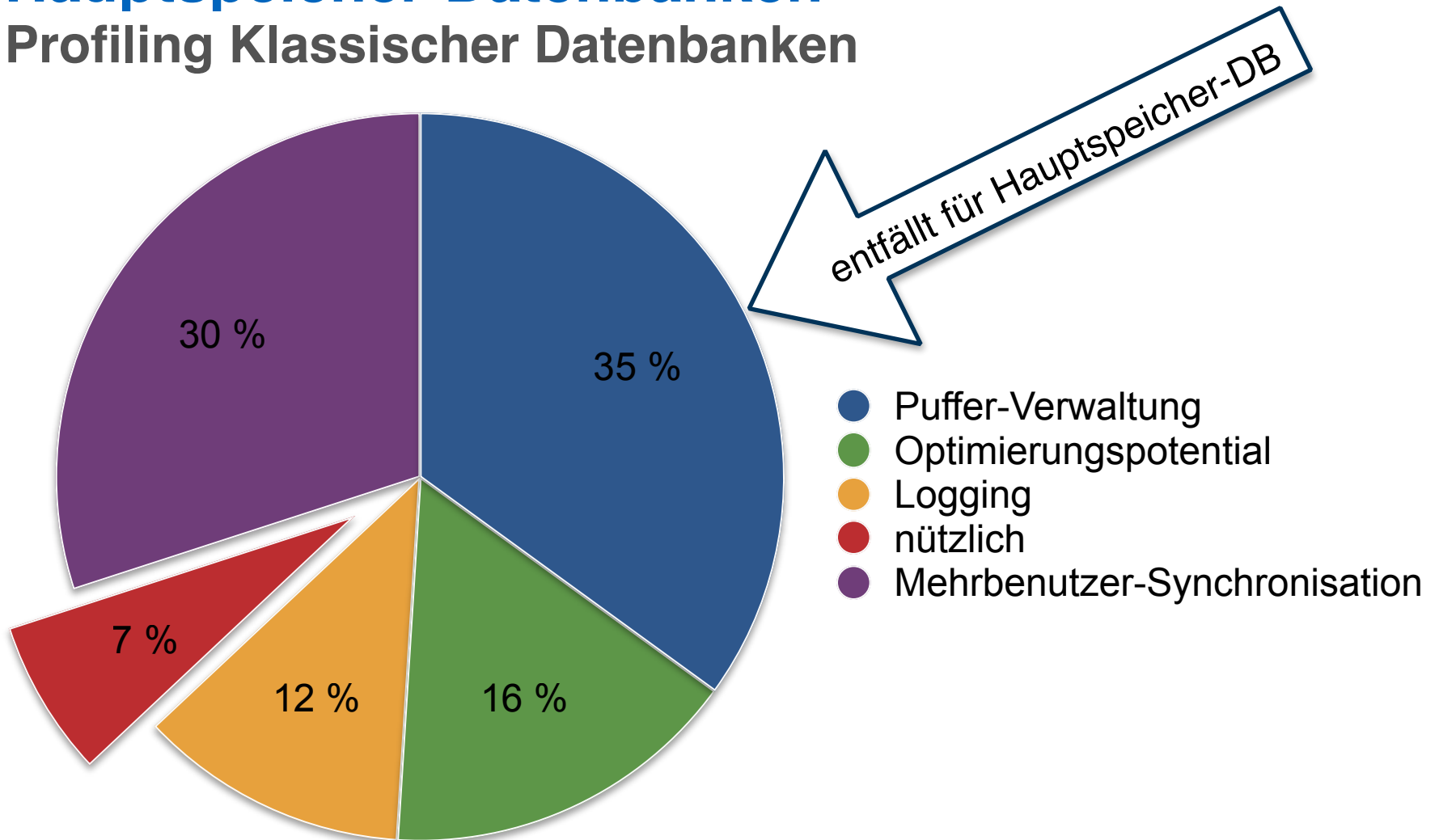
Hauptspeicher-Datenbanken

Einsatz



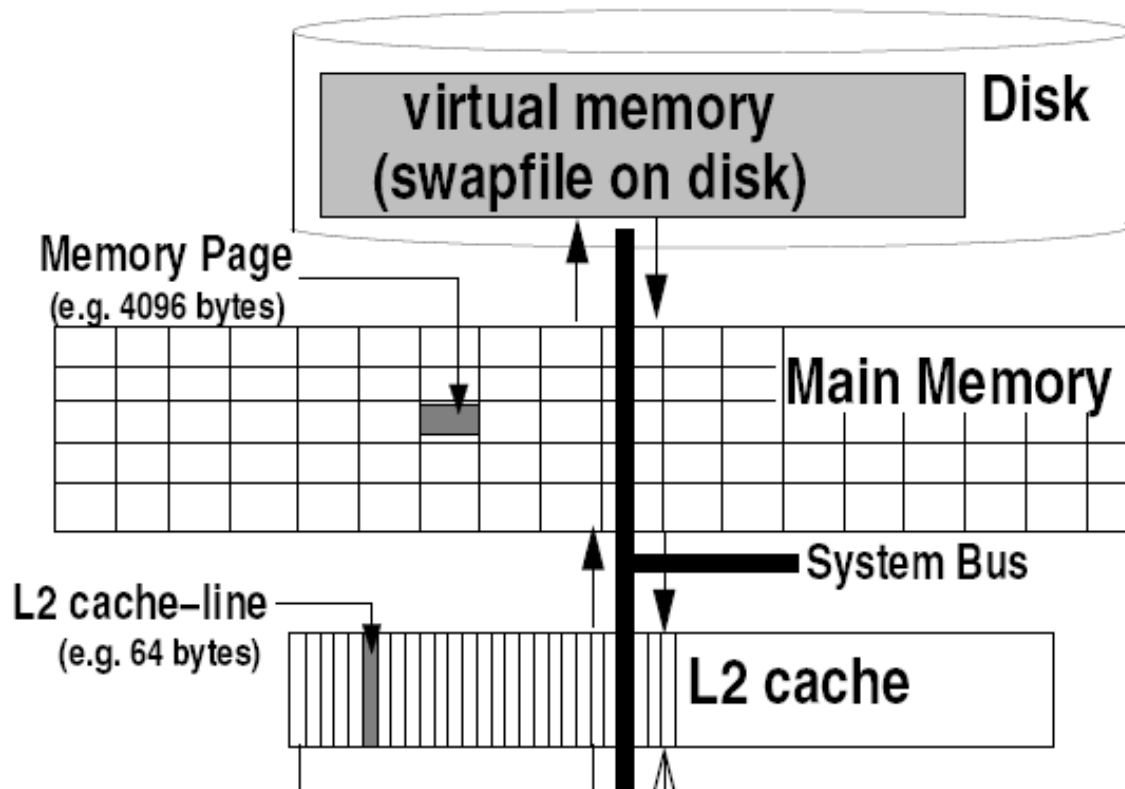
Hauptspeicher-Datenbanken

Profiling Klassischer Datenbanken



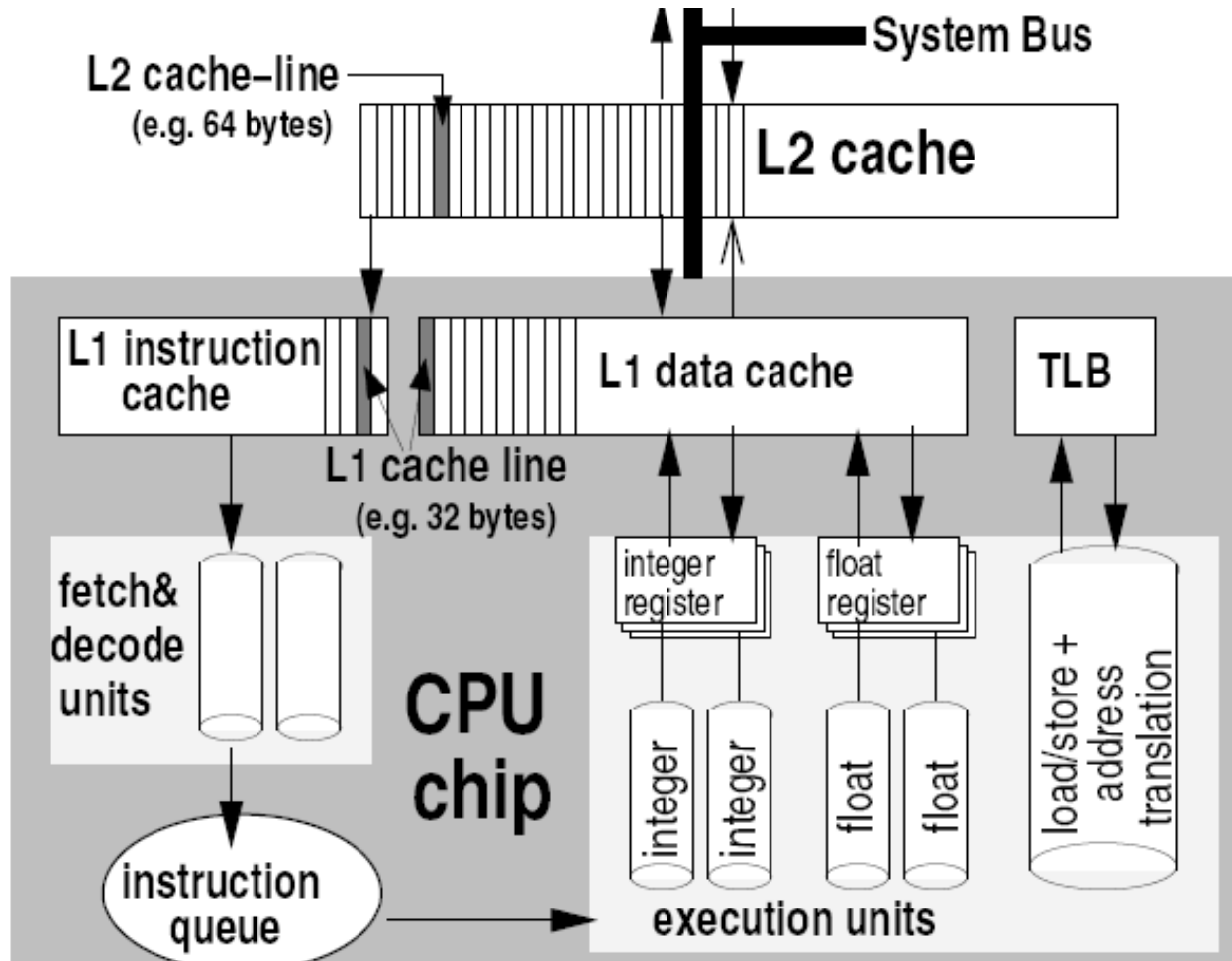
Hauptspeicher-Datenbanken

Speicherhierarchie



Hauptspeicher-Datenbanken

Speicherhierarchie





Aufgabe 1

HyPer schafft 120.000 Transaktionen pro Sekunde. Pro Transaktion werden 120 Byte in die Log geschrieben. Berechnen Sie den benötigten Durchsatz zum Schreiben der Log.

Die Datenbank läuft für einen Monat und stürzt dann ab. Es wurde kein Snapshot erstellt. Berechnen Sie die Recoveryzeit. Gehen Sie davon aus, dass die Recovery durch die Festplatte limitiert ist (100 MiB / s). Wieviel Log Einträge werden pro Sekunde reconvert?

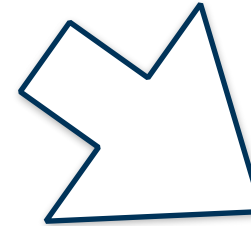
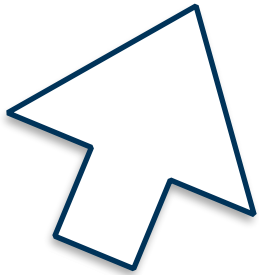


Hauptspeicher-Datenbanken

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

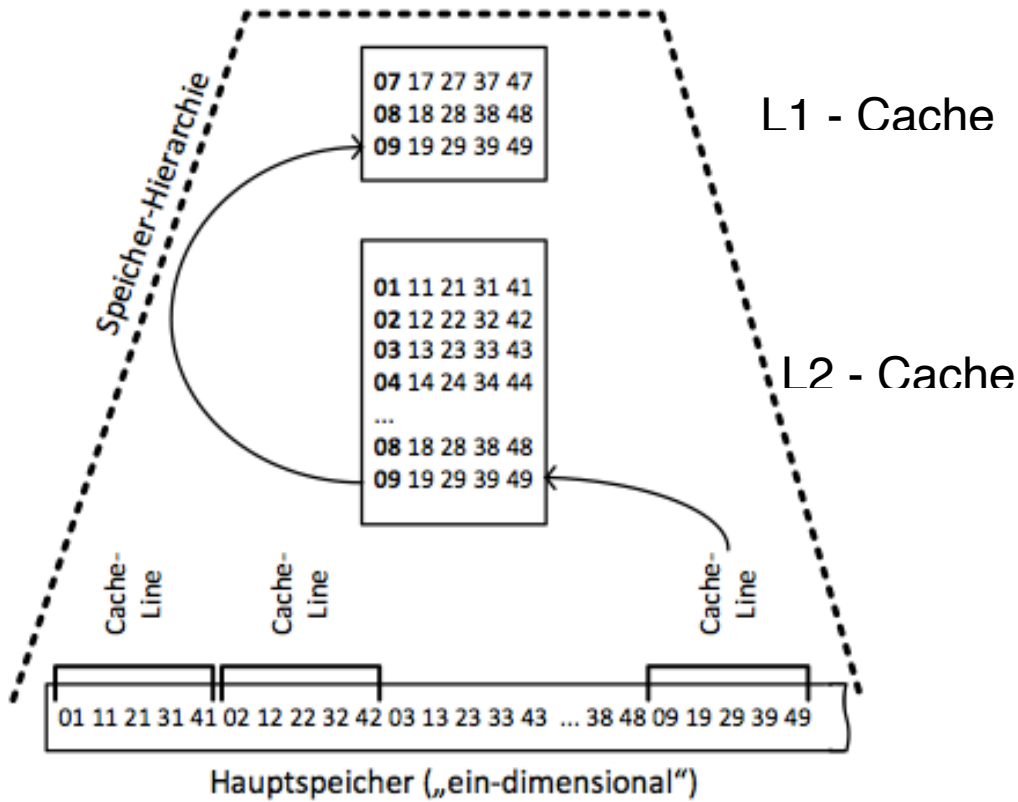
Name	MatrNr	Semester	Fach	Nebenfach
Alex	362148	6	Info	Medizin
Max	362139	6	Info	Physik
David	361299	10	Info	MaschBau
Johannes	362033	8	Info	Mathe
Andre	362101	10	Info	Mathe



Hauptspeicher-Datenbanken

Row-Store

**select sum(A)
from R**



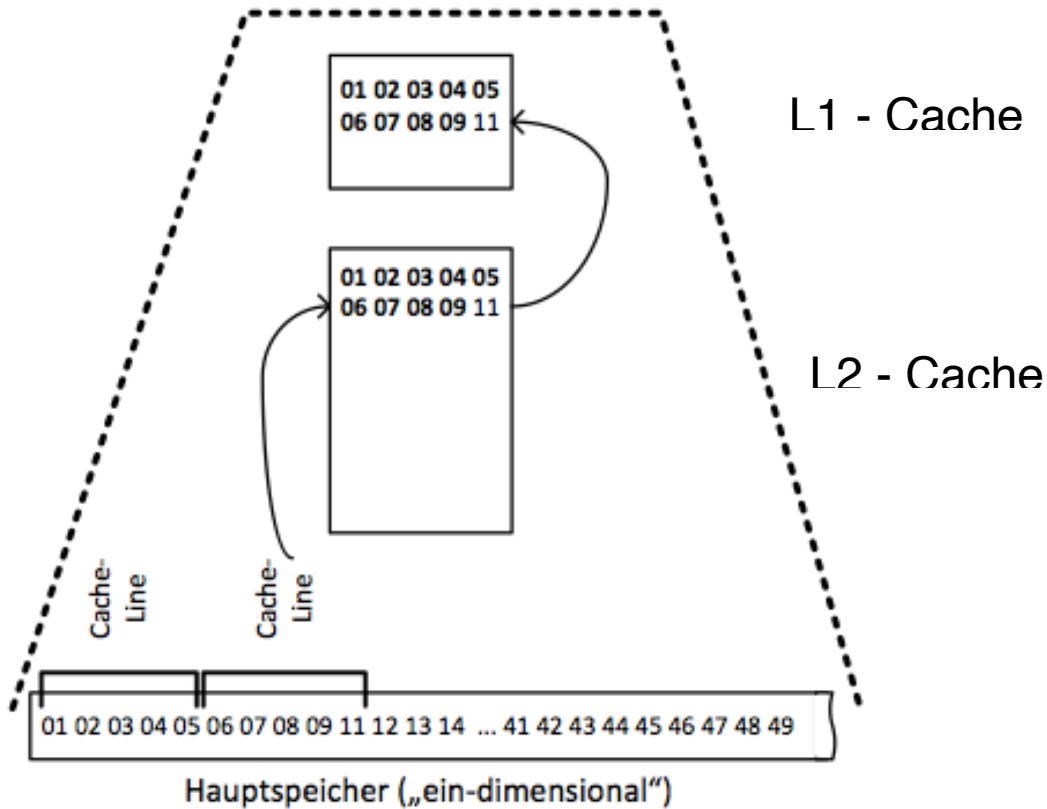
Speicherstruktur

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

**select sum(A)
from R**



Speicherstruktur

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



Aufgabe 2

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*



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
...				

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
...

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

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:

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

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
...				

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
...

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

```
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
...				

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
...

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

```
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:

$$\#Cachelines = \lceil 32B/64B \rceil + \lceil 3B/64B \rceil = 2$$

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
...				

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
...

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(...);

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:

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



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 3

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 4

Rekonstruieren Sie die ursprüngliche SQL-Anfrage aus dem folgenden (Pseude-)Code eines codegenerierenden Datenbanksystems. Welche Art von Join wurde benutzt? Handelt es sich um Column- oder Row-Store?

```
class Student { int matrnr; std::string name; int semester; }
class Hoeren { int matrnr; int vorlnr; }
class Result { int vorlnr, a; }

std::vector<Result> compute(std::vector<Student> ses, std::vector<Hoeren> hs){
    std::unordered_multimap<int, Hoeren*, Studenten*> h_map;
    std::unordered_map<int, Studenten*> s_map;
    for(auto h: hes)
        h_map.insert(std::pair<int, Hoeren*>(h.matrnr, &h, nullptr))
    for(auto s: ses)
        s_map.insert(std::pair<int, Studenten*>(s.matrnr, &s))

    // Group by h.vorlnr; avg(s.semester) = sum(s.semester)/count(*)
    std::unordered_map<int, int> count_map;
    std::unordered_map<int, int> sum_map;
    for(auto h: hes){
        count_map.insert(std::pair<int, int>(h.vorlnr, 0))
        sum_map.insert(std::pair<int, int>(h.vorlnr, 0))
    }
    for(auto h: h_map){
        sum_map[h->second->vorlnr] += s_map[h->first]->semester
        count_map[h->second->vorlnr]++;
    }
    std::vector<Result> res;
    for(auto r: sum_map)
        res.push_back(r->first, r->second/count_map[r->first]);
    return res;
}
```



Aufgabe 5

Führen Sie die folgenden Abfragen in der Spark-Shell aus. Als Grundlage für die Abfragen dient das TPC-H Schema. Laden Sie dazu die TPC-H Daten wie in der Vorlesung gezeigt in die Spark-Shell.

- (a) Ermitteln Sie pro Marktsegment die Anzahl der Bestellungen in 1997.
- (b) Ermitteln Sie die Zahl der Kunden und Lieferanten pro Land.
- (c) Ermitteln Sie die Stückzahlen der verschiedenen Bauteile in Deutschland.
- (d) Ermitteln Sie, welche Kunden kein *goldenrod lavender spring chocolate lace* bestellt haben.



Aufgabe 5

Load all tables

- Load the tpch.scala file into your Spark Shell:
 - **Download tpch.scala:** <https://tinyurl.com/3383aa4n>

```
wget https://tinyurl.com/3383aa4n/download/tpch.scala
```
 - **Update DATA_PATH variable in tpch.scala:**

```
val DATA_PATH = /the/path/to/your/tpc-h/data
```
 - **Load the script into your shell:**
 - Option 1: Load the script into running Spark Shell:

```
:load /path/to/tpch.scala
```
 - Option 2: Restart Spark:

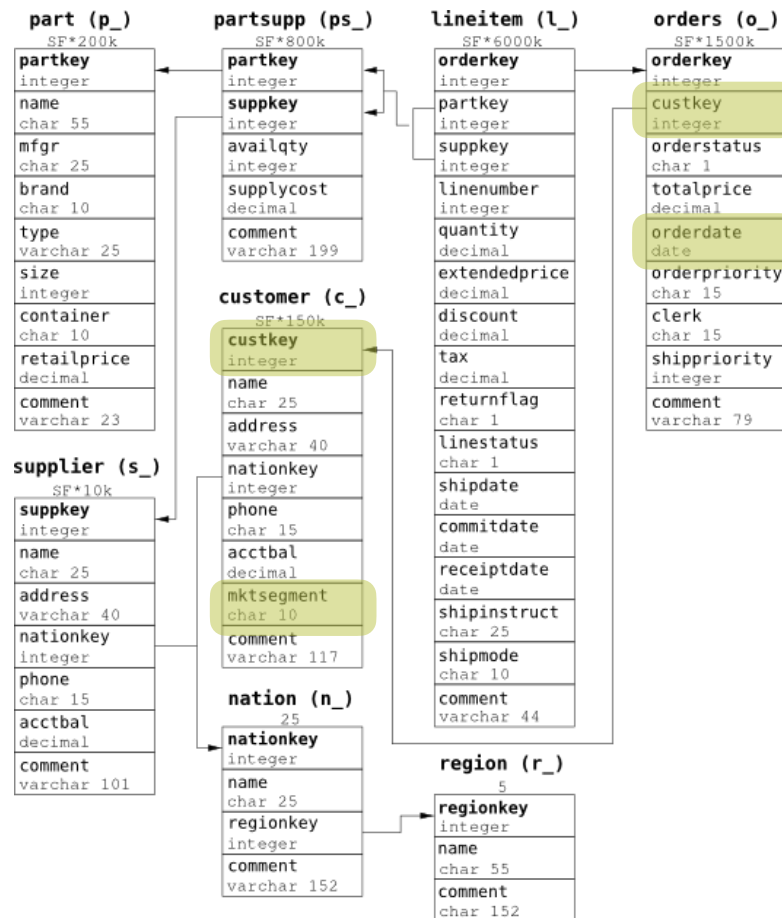
```
./bin/spark-shell -l /path/to/tpch.scala
```



Aufgabe 5

TPC-H Schema

(a) Ermitteln Sie pro Marktsegment die Anzahl der Bestellungen in 1997.

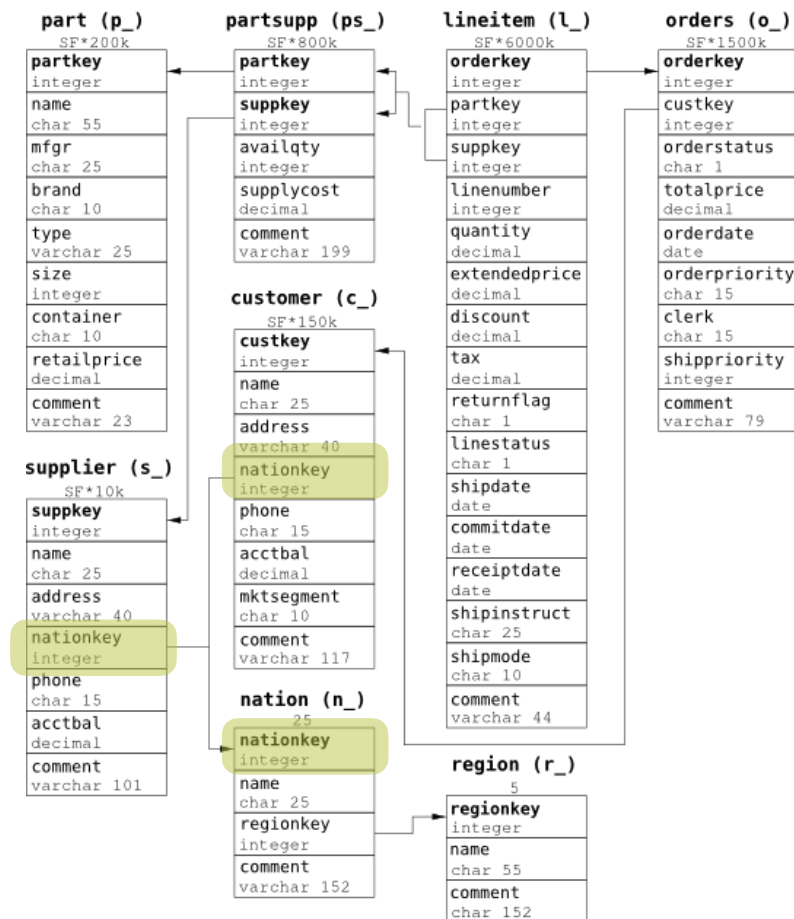




Aufgabe 5

TPC-H Schema

(b) Ermitteln Sie die Zahl der Kunden und Lieferanten pro Land.

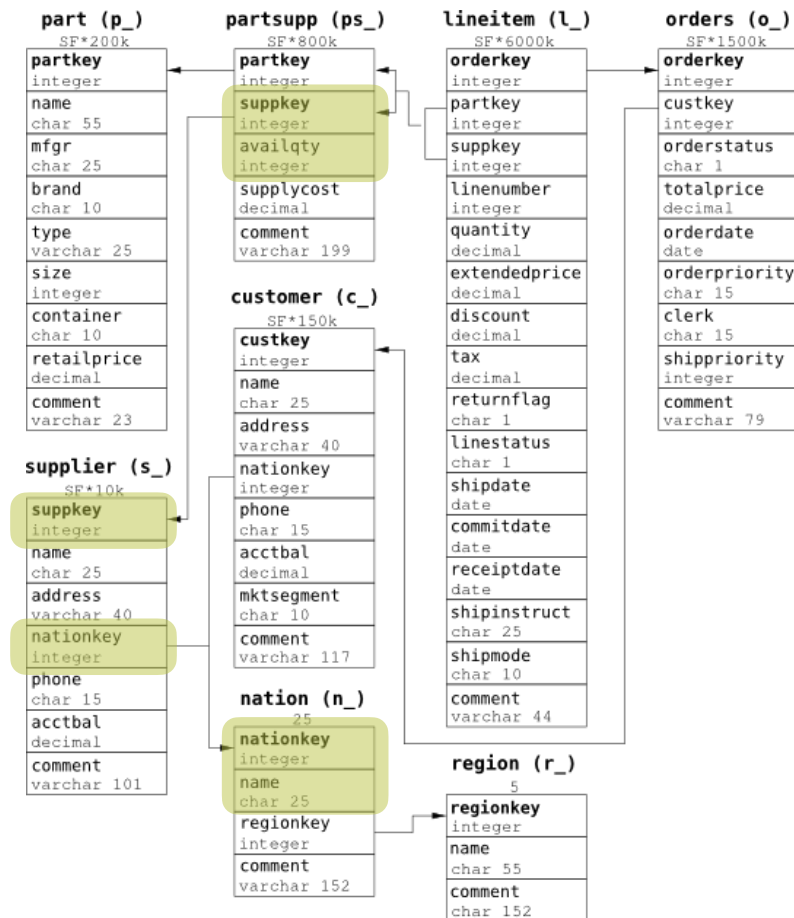




Aufgabe 5

TPC-H Schema

(c) Ermitteln Sie die Stückzahlen der verschiedenen Bauteile in Deutschland.





Aufgabe 5

TPC-H Schema

(d) Ermitteln Sie, welche Kunden kein *goldenrod lavender spring chocolate lace* bestellt haben.





Aufgabe 6

TPC-H Schema

Führen Sie die folgenden Abfragen in der Spark-Shell aus. Als Grundlage für die Abfragen dient das TPC-H Schema. Laden Sie dazu die TPC-H Daten wie in der Vorlesung gezeigt in die Spark-Shell.

- (a) Laden Sie die `region.tbl` Datei als DataFrame Objekt in die Spark-Shell.
- (b) Ermitteln Sie die Namen aller Regionen.
- (c) Ermitteln Sie die Zahl der Länder die nicht in Europa liegen.
- (d) Ermitteln Sie die größte Bestellung aus dem Jahr 1996.
- (e) Ermitteln Sie welcher europäische Kunde im Jahr 1996 am meisten Geld ausgegeben hat.
- (f) Ermitteln Sie welche Unternehmen keine Kunden in Europa haben.



Fragen?