

Big-Data-Technologien

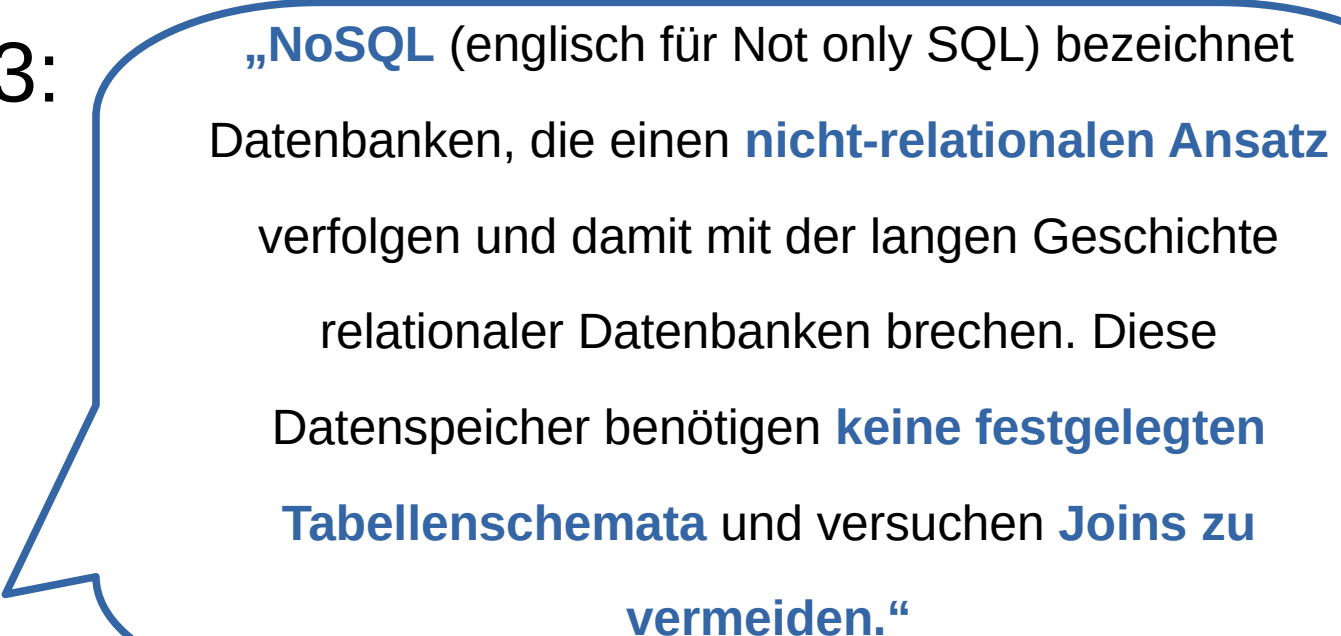
Kapitel 8: NoSQL – Einführung

Hochschule Trier
Prof. Dr. Christoph Schmitz

Warum „NoSQL“?



Warum „NoSQL“?

- Klärung 1: „**Not only SQL**“
- Klärung 2: **Es geht nicht um SQL!**
- Klärung 3:  „**NoSQL** (englisch für Not only SQL) bezeichnet Datenbanken, die einen **nicht-relationalen Ansatz** verfolgen und damit mit der langen Geschichte relationaler Datenbanken brechen. Diese Datenspeicher benötigen **keine festgelegten Tabellenschemata** und versuchen **Joins zu vermeiden.**“

Crashkurs: Relationale Datenbank-Management-Systeme (RDBMS)

- Relati
- Frem
- Relati
- Logis
- Trans

Information Retrieval

CACM Vol. 13(6), 1970

A Relational Model of Data for Large Shared Data Banks

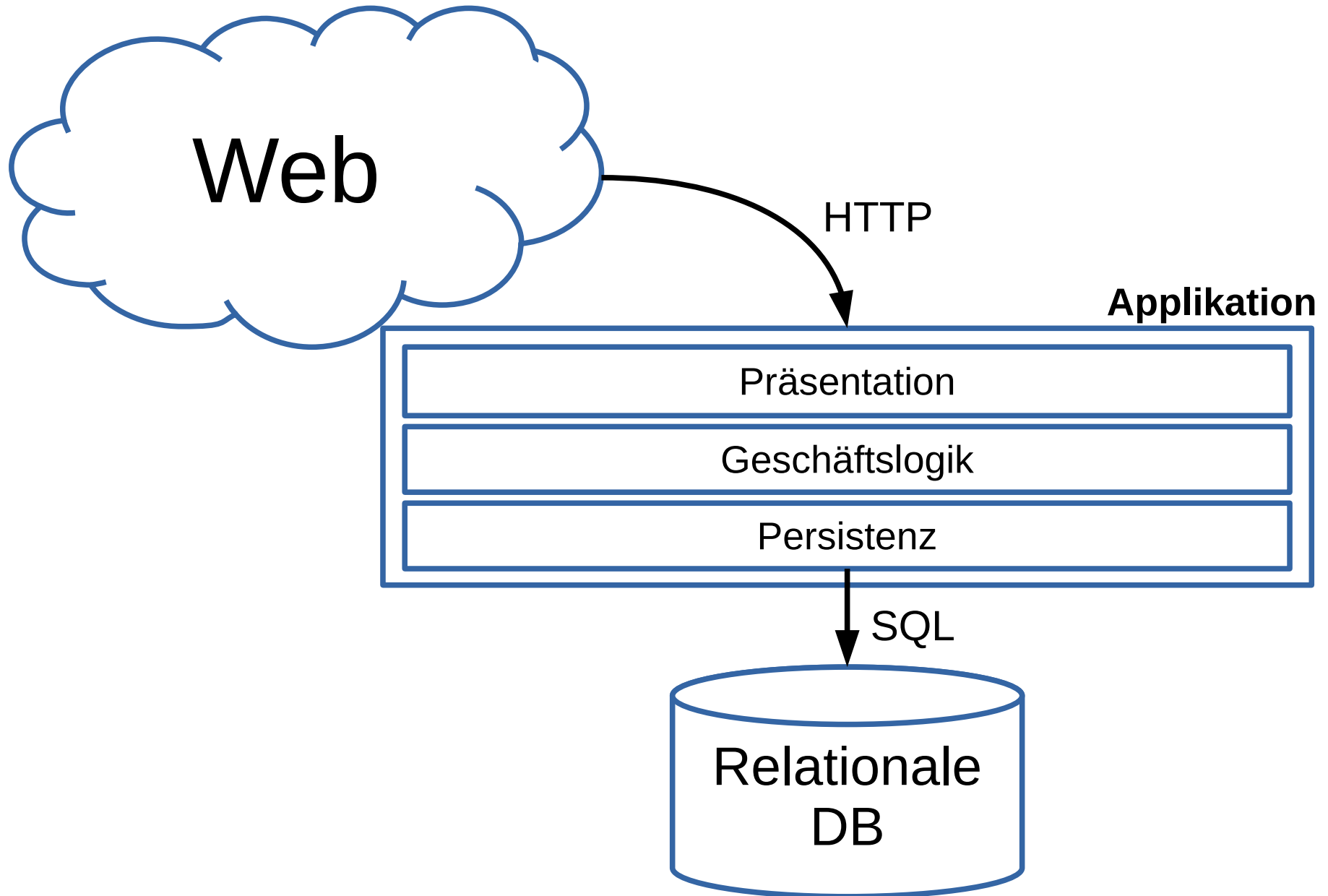
E. F. CODD

IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain

me
üller
hmidt

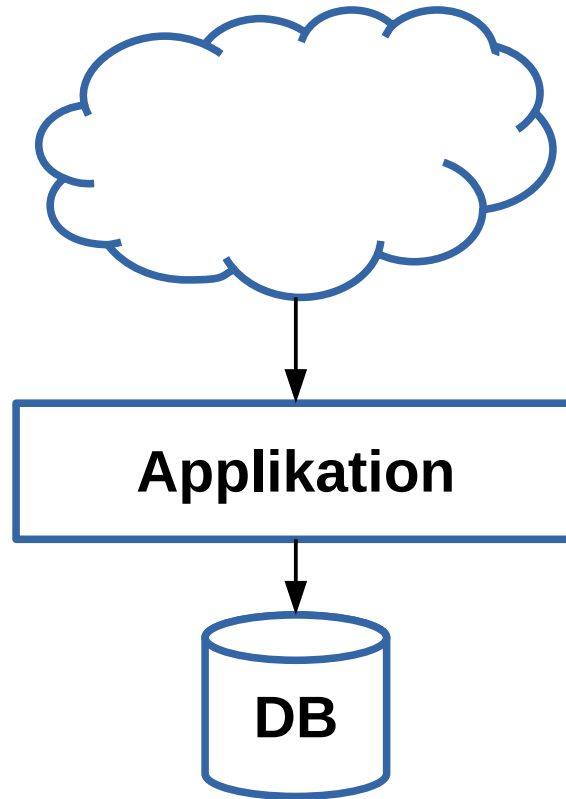
Historie: State of the Art, ca. 2000



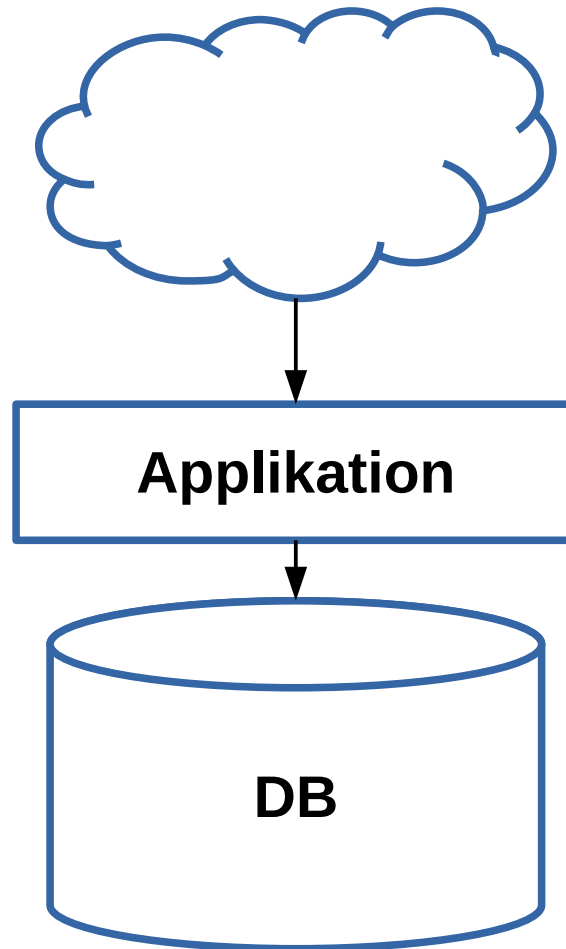
Was ist das Problem?

- RDBMS sind bis heute der Standard, aber...
- ... Probleme mit
 - **Skalierbarkeit**
 - **Datenmodell und Flexibilität**
 - **„Impedance Mismatch“**

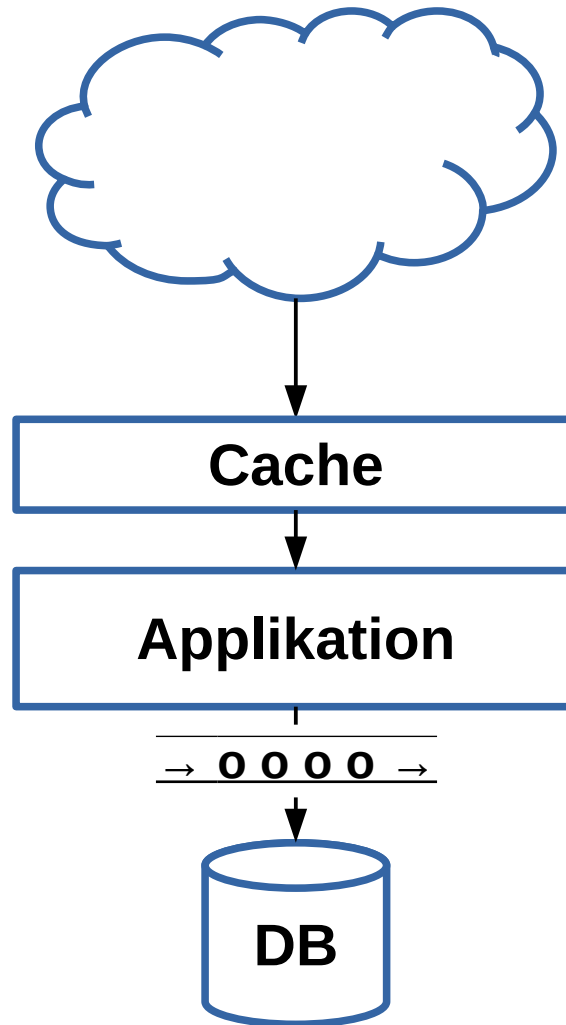
Skalierbarkeit



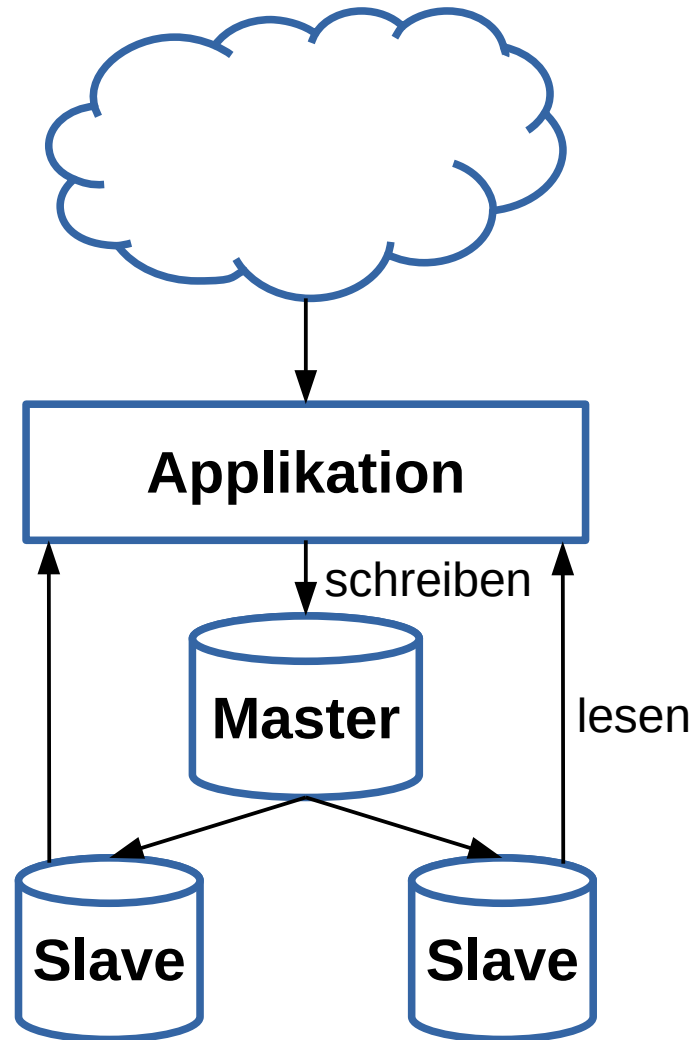
Skalierbarkeit: Scale Up



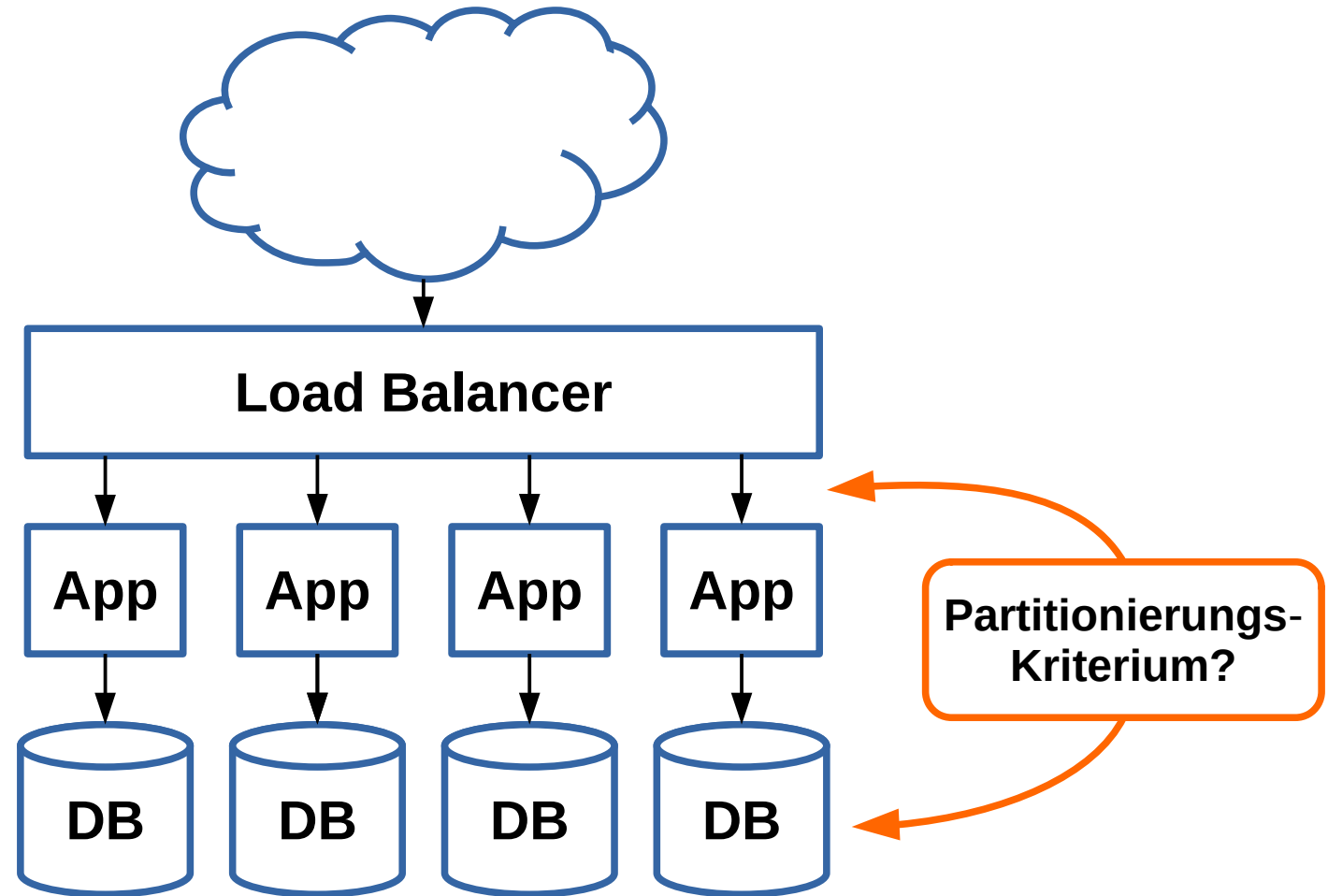
Skalierbarkeit: Caching, Queues



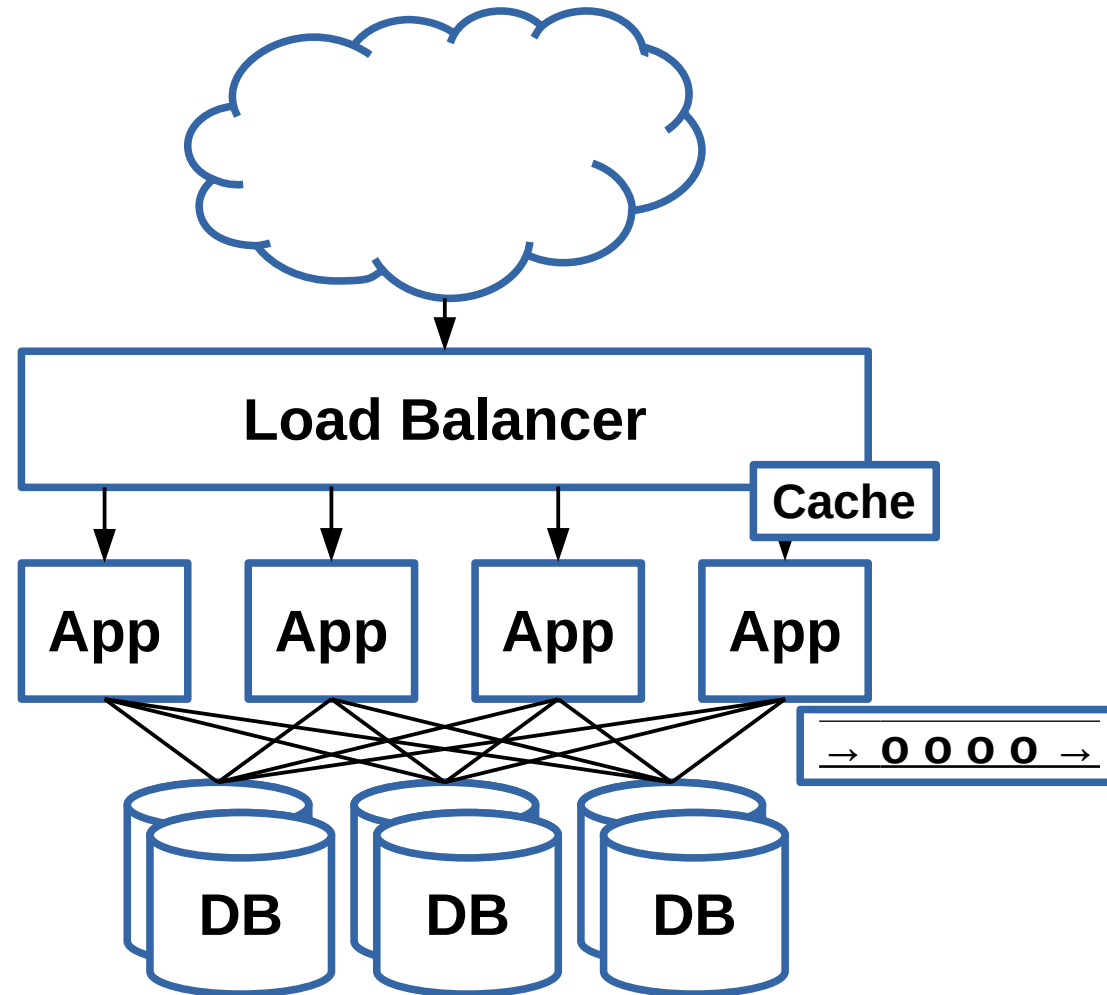
Skalierbarkeit: Master-Slave-Replikation



Skalierbarkeit: Sharding (einfach)



Skalierbarkeit: Sharding (realistisch)

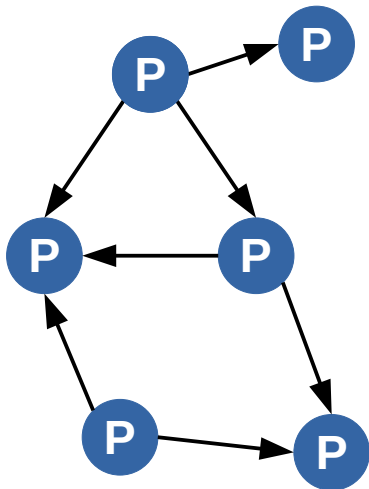


Impedance Mismatch

```
class Person {  
    String name;  
    int age;  
    Set<Person> friends;  
    Set<Person> children;  
    Person father;  
    Person mother;  
}
```

```
CREATE TABLE person (  
    id INTEGER ...,  
    name VARCHAR(100),  
    ...  
)
```

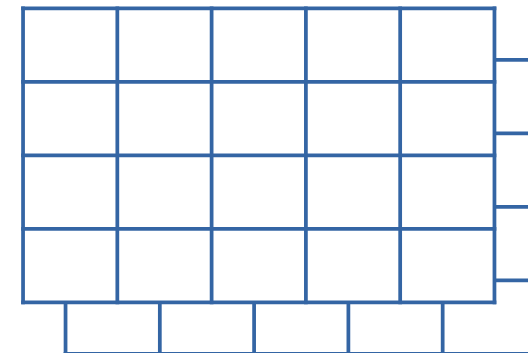
```
CREATE TABLE friend_of (  
    id1 INTEGER,  
    id2 INTEGER,  
    FOREIGN KEY(id1) ...,  
    ...  
)
```



Objekte



- JPA
- Hibernate
- LINQ



Relationen

Flexibilität

- RDBMS verlangen **fixes Schema**
- Was tun bei **Änderungen**?
 - Ansatz 0: **Schema-Missbrauch**
 - Ansatz 1: **Migrationsprojekt**
 - Ansatz 2: **Adapterschichten**
 - Ansatz 3: **Metamodellierung**



```
CREATE TABLE fact (  
  subject INTEGER,  
  predicate INTEGER,  
  object INTEGER  
)
```

SUBJECT	PREDICATE	OBJECT
kunde-1	erhält	rechnung-34
rechnung-34	enthält	produkt-17
produkt-17	kostet	123,50

- Ansatz 4: **Microservices**

Was bietet also NoSQL?

- **Skalierbarkeit**
 - Durchsatz/Latenz
 - Vereinfachung
 - Verteilung
- **Flexible Datenmodelle**
 - komplexere Strukturen
 - Erweiterbarkeit

Welche Kompromisse muss ich machen?

- **Fehlende Eigenschaften** aus RDBMS
 - Joins
 - Transaktionen
 - Sicherheitsaspekte
 - **Standardisierung**
- **Kompromisse** durch Verteilung: **CAP-Theorem**
 - Konsistenz
 - Verfügbarkeit
 - Partitionstoleranz

Ausblick: Typologie der NoSQL-Systeme

- Key-Value Stores
- Dokumentenorientierte Datenbanken
- Wide Column Stores
- Spaltenorientierte Datenbanken
- Graphdatenbanken

Key-Value Stores

- **Beschreibung**

- Speichern zu jedem Schlüssel (Key) einen Wert
- oft im RAM → extrem hoher Durchsatz und geringe Latenz

- **Einsatzgebiete**

- Auslieferungssysteme
- Caches

- **Vertreter**

- Memcache
- **Redis**
- Voldemort
- Riak
- Aerospike

K	V
uid1	{}
uid2	{a, b, c ,d}
uid3	{c, e, h, i, j}

Dokumentenorientierte Datenbanken

- **Beschreibung**

- Speichern semistrukturierte Dokumente
- Flexible Abfragemöglichkeiten

- **Einsatzgebiete**

- Volltextsuche
- semistrukturierte Daten

- **Vertreter**

- **MongoDB**
- Elasticsearch
- CouchDB
- Solr

```
{  "id": "blogentry1",
  "title": "Big Data",
  "text": "...",
  "tags": [
    "bigdata",
    "document",
    "database"
  ],
  "comments": {
    {  "id": "c11",
      "text": "..."},
    {  "id": "c12",
      "text": "..."}
  }
}
```

Wide Column Stores

- **Beschreibung**

- Zeilen mit flexibler Anzahl von Spalten
- Spaltennamen kodieren Nutzdaten

- **Einsatzgebiete**

- Webanwendungen
- Flexible Datenhaltung

- **Vertreter**

- **Cassandra**
- HBase
- Accumulo

id	title	sports	hobby	fun
page1	Soccer	1	1	1

id	title	bigdata	lecture
page2	BigData	1	1

Spaltenorientierte Datenbanken

- **Beschreibung**

- relationales Datenmodell
- Speicherung spalten- statt zeilenweise

- **Einsatzgebiete**

- Analysen
- Business Intelligence

- **Vertreter**

- SAP HANA
- Sybase IQ
- Exasol
- (Oracle/DB2/SQL Server)

Name	City	Gender
Alice	TR	f
Bob	TR	m
Charlie	SB	m
...	TR	m
...	KO	f
...	KO	f
...	SB	m
...	TR	f

Graphdatenbanken

- **Beschreibung**
 - Datenmodell sind Property-Graphen
 - Abfragesprachen für Graphen
- **Einsatzgebiete**
 - flexible, erweiterbare Datenhaltung
 - semistrukturierte Daten
- **Vertreter**
 - Neo4j
 - Titan
 - OrientDB

