Big Data

Datenmodelle

Andreas Scheibenpflug

Relationales Datenmodell

- Tabellen, Spalten und Zeilen

E-Mail	Vorname	Nachname	Notiz
max@domain.com	Max	Mustermann	Test
franz@email.com	Franz	Mayr	Kartoffel
...	...	...	...

- Primärschlüssel, Fremdschlüssel
- Beziehungen: 1:1, 1:n, n:m

Wide Column Stores (1/2)

- Ähnlich zu relationalen Modellen
  - Tabellen, Spalten, Zeilen
- Physisch wird spaltenweise gespeichert
- Aber
  - Keine Joins. Ersatz: Redundanz, Denormalisierung
  - Einschränkungen bei Selects und Aggregationen (Group by, Sorting, Sum,...)
- Potenziell viele Spalten möglich
- Sparse columns, Column Families
- Zelle: Triplet aus Zeile, Spalte, Version
- Je nach DB Schema-frei

Wide Column Stores (2/2)

- Bei relationalen Modellen wird von den Daten ausgegangen
  - Daten → Tabellen → Relationen → Joins → Anwendung
- Bei Wide Column Stores wird bei der Modellierung von Anwendungsabläufen ausgegangen
  - „Query-driven approach“: Modellierung der Daten nach den von der Anwendung benötigten Abfragen
  - Anwendung → Query → Datenmodell

Key Value Stores

- Speicher von Tuples Key → Value
- Schema-frei, Value ist ein Blob

Key	Value
max@domain.com	Max Mustermann
franz@email.com	Franz Mayr

- Oft zusätzliche Unterstützung von anderen Datentypen
  - Listen, Sets, Maps, Sortierte Sets, Documents, ...

Document DBs (1/2)

- Speichern strukturierter Dokumente
  - Meist im XML oder JSON Format
- Keine n:m Beziehungen
- Indizes, Aggregationen möglich

```
{
     email: "max@domain.com",
     vorname: "Max",
     nachname: "Mustermann",
     notiz: "test",
}
```

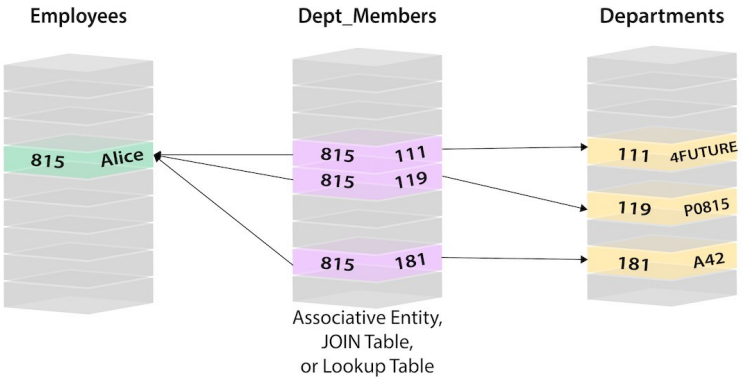
Document DBs (2/2)

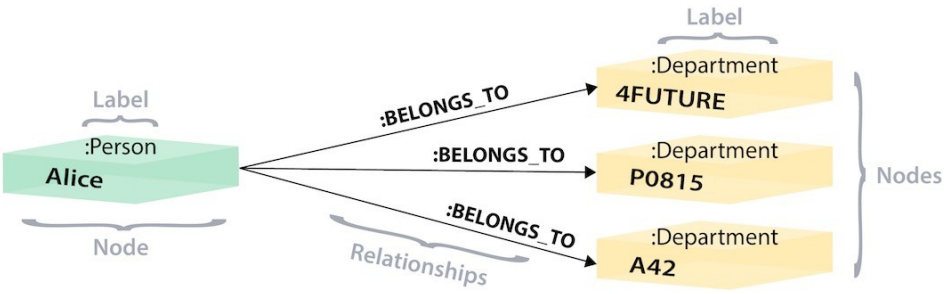
Ähnliche Dokumente werden zu Collections zusammengefasst

```
{
     _id: "max@domain.com",
     vorname: "Max",
     nachname: "Mustermann",
     address: {
               strasse: "Musterstraße 3",
               plz: "Musterstadt"
     }
}
{
     _id: "franz@email.com",
     vorname: …
     ...
}
```

Graph DBs

- Knoten, Kanten, Attribute



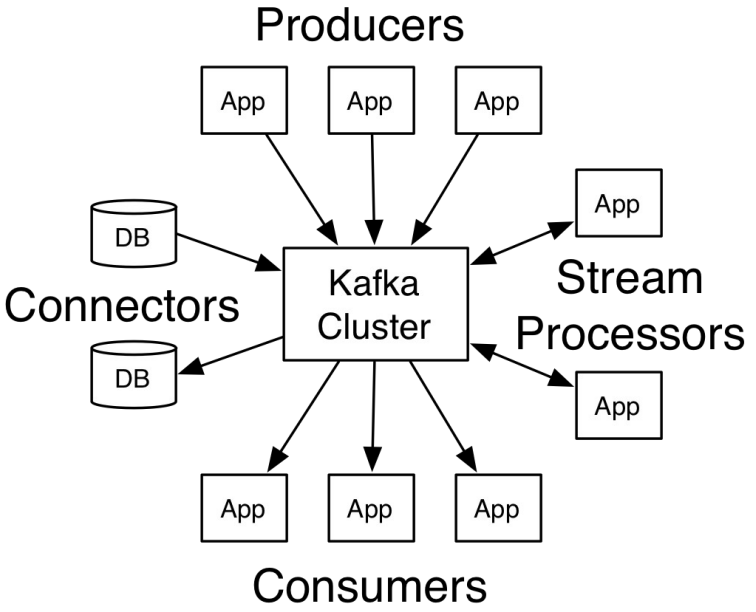


Grafiken: Neo4j, https://neo4j.com/developer/graph-db-vs-rdbms/

- Beispiel: Social Graph (Facebook,…)

Datenstrom orientiert

- Ähnlich zu Message Broker (Publish/Subscribe)
- Message Broker + Processing API + Storage
- Echtzeitverarbeitung von Datenströmen
  - Sensordaten, IoT
  - LinkedIn: User Activity Tracking
  - Uber: Passenger/Driver matching



Grafik: Kafka, https://kafka.apache.org/intro

Wann was?

- Eine Frage
  - der Struktur/Eigenschaften der Daten
  - der gewünschten Performance des Systems
  - den Anforderungen an Konsistenzverhalten
- In vielen Systemen ein Mix
  - Microservice Architekturen erleichtern es für Services unterschiedliche Datenbanken zu verwenden