

Relationale  
Datenbanken

V S

NoSQL-  
Datenbanken

# Agenda

- Relationale DBs
  - Aufbau relationaler Datenbanken
  - ACID-Eigenschaften
- NoSQL DBs
  - Ansätze von NoSQL
  - BASE Modell
  - CAP-Theorem
- Vergleich beider Modelle

# Relationale Datenbanken

- Das auf dem Markt am weitesten verbreitete Modell
- Architektur basiert auf das Entity-Relationship-Modell (ERM)
  - Entwickelt von Edgar F. Codd im Jahr 1971
  - Jede Anfrage lässt sich in relationale Algebra übersetzen
- Setzt auf Vertikale Skalierung
- SQL wurde als Sprache für Datenbankabfragen entwickelt
  - *Structured Query Language*
  - Mittlerweile ein inoffizieller Standard
  - RDBS werden oft auch als SQL-Datenbanken bezeichnet

# Aufbau von RDBS

- Eine Tabelle (Beziehung) besteht aus mehreren Attributen
  - Organisation in Zeilen (Datensatz) und Spalten (Attribute)
  - Komplette Tabelle ist eine Ansammlung von Tupeln
  - Verwendung von Schlüssel zur Identifizierung eines Datensatzes

Tabelle (Relation)

Kundendaten

Kd-Nr	Nname	Vname	Strasse	Plz	Ort
0001	Schneider	Thomas	Hauptstr. 2	50969	Köln
0002	Maier	Wolfgang	Gartenstr. 1	77933	Lahr

Attribut

Primärschlüssel

Datenfeld

Datensatz (Tupel)

# ACID

- RDB-Transaktionen werden durch 4 Eigenschaften definiert
- Atomicity, Consistency, Isolation und Durability (kurz. ACID)
  - Atomarität      Definiert alle Elemente, die eine vollständige Datenbanktransaktion ausmachen
  - Konsistenz      Regeln, dass Datensätze nach einer Transaktionen den korrekten Zustand bewahren
  - Isolation      Auswirkungen einer Transaktionen bleibt für andere unsichtbar, bis sie durchgeführt wurde
  - Dauerhaftigkeit      Datenänderungen sind dauerhaft, sobald die Transaktion durchgeführt wurde

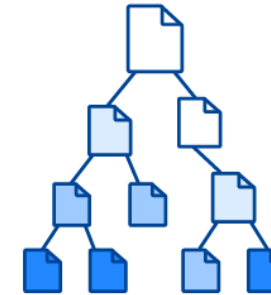
# NoSQL Datenbanken

- Steht für *Not only SQL*
  - Setzt auf alternative Modelle im Vergleich zu RDBS
  - Überwindet Grenzen der relationalen Datenbanken
  - Eine Mischung ist dennoch möglich
- Flexibel, kein Tabellenschema
  - Flexible Einarbeitung von Daten
  - Verarbeitung unstrukturierter Daten (nicht möglich für RDBS)
- Setzt auf Horizontale Skalierung

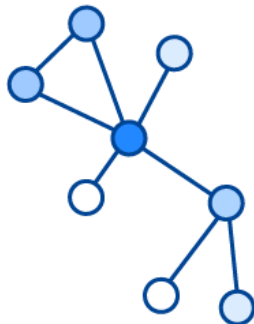
# NoSQL Ansätze

- Dokument
  - Daten in Dokumenten variierender Länge
  - Zuordnung von Attributen
  - Geeignet für CMS und Blogs
  - JSON als gängiges Datenformat

Document



Graph



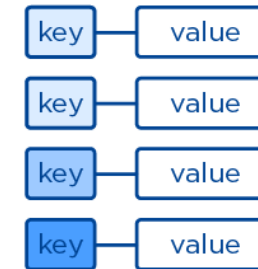
- Graph
  - Beziehungen durch Knoten und Kanten
  - Organisation durch Beziehungsgeflecht
  - Anwendung im Bereich Social Media
    - Beziehung zwischen Followern

# NoSQL Ansätze

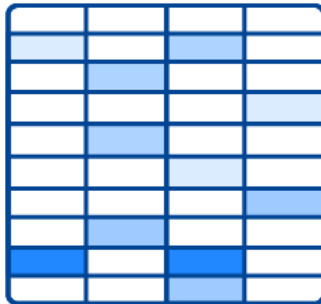
- Key-Value

- Speicherung als Schlüssel-Wert-Paar
- Jeder Wert besitzt einen eindeutigen Schlüssel
- Schlüssel sind immer eindeutig

## Key-Value



## Wide-column



- Spalten

- Speichern Daten in Spalten
- Kürzere Leseprozesse und Höhere Leistungsfähigkeit
- Anwendung im Data-Mining und Analyseprogrammen



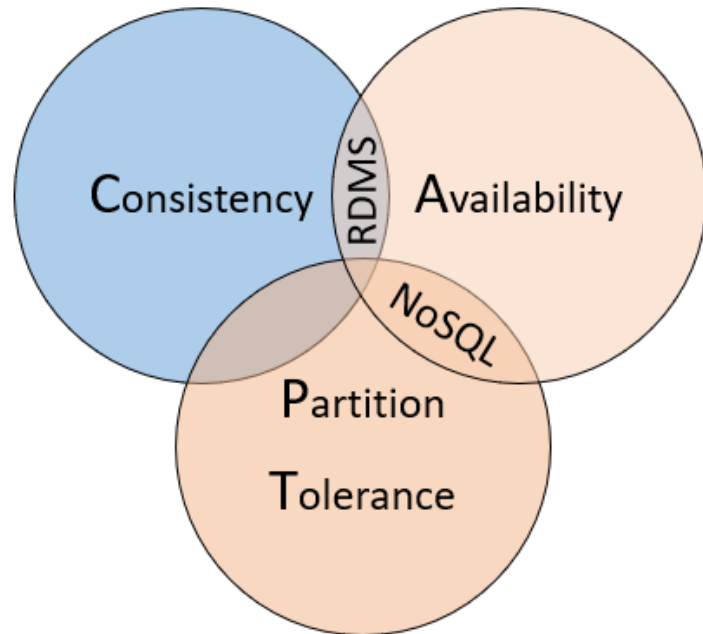
# BASE Modell

- NoSQL Variante des ACID-Modell
- Charakterisiert das flexible Verständnis von NoSQL-Systemen
- **B**asic **A**vailability, **S**oft State und **E**ventual Consistency
  - Basic Availability      Operationen sind solange Verfügbar wie möglich, aber nicht konsistent.
  - Soft State              Zustand des Systems kann sich über Zeit ändern
  - Eventual Consistency    System wird konsistent, wenn nach gewisser Zeit kein Input erfolgt
- BASE gibt die absolute Konsistenz auf

# CAP-Theorem

- Anderer Ansatz für ACID und BASE
- **C**onsistency, **A**vailability und **P**artition Tolerance
  - Konsistenz  
Jeder Knoten antwortet mit den neuesten Daten, auch wenn das System die Anforderung blockieren muss, bis alle Daten aktualisiert werden.
  - Verfügbarkeit  
Jeder Knoten gibt eine sofortige Antwort, auch wenn es sich nicht um die neuesten Daten handelt.
  - Partitionstoleranz  
Stellt sicher, dass das System weiterhin funktioniert, auch wenn ein Datenknoten ausfällt.

# CAP-Theorem



- Anders als bei ACID kann NoSQL immer nur 2 von 3 Eigenschaften garantieren
- RDMS bieten i.d.R. keine Partitionstoleranz
  - Werden von einem Server bereitgestellt
  - Vertikale Skalierung
- NoSQL bietet Partitionstoleranz
  - Möglichkeit über Server-Cluster
  - Horizontale Skalierung
- Wird auch auf PACLEC erweitert

# Vergleich beider Systeme

	SQL-Datenbank	NoSQL-Datenbank
Art	DB für alles	Verschiedene Modelle
Datenspeicherung	In Datensätzen in einer Tabelle und Attributen zugeordnet	Dokumente, Key-Values, Graphen oder Spalten
Schemata	Datentypen und Struktur im vorhinein festgelegt	Flexibel, keine vorherige Konvertierung nötig
Skalierung	Vertikale Skalierung	Horizontale Skalierung
Modell	ACID-Modell	BASE-Modell
Leistung	Abfall bei großen Datenmengen	Höhere Leistungsstärke
API	SQL	Objektbasierte APIs

# Fazit

- Welcher System besser ist hängt von der Anwendung ab
  - RDBS bei strukturierten Daten
  - NoSQL bei unstrukturierten Daten
- Skalierbarkeit eines Systems spielt eine Rolle
  - Vertikale vs. Horizontale Skalierung bei sehr großen Datenbanken