

Aaron Schreiegg, Filip Schauer, Emil Silber, Axel Csomany, Simon Pesut

# Übersicht

---

- "Mehrere Datenbanken in einem"
- Mehrere Datenmodelle
  - Document Model
  - Graph Model
  - Key-Value Model
  - Relational Model
- Verschiedene Data Types

## Datenmodelle

---

### Document Model

- JSON-ähnliche Dokumente
- Semi-Strukturierte Daten
- Beispiel:

```
{  
  "id": 1,  
  "name": "Laptop",  
  "price": 1200  
}
```

### Graph Model

- Nodes und Edges
- Für Relationship-based Data oder Recommendation Systems

### Key-Value Model

- Einfaches Key-Value-Pair
- Schnelle Datenabfrage
- Beispiel:

```
"order_123": {  
  "total": 500,  
  "items": ["item1", "item2"]  
}
```

### Relation Model

- Klassische Tabellenstruktur (Columns, Rows)
- Ideal für Transactional Systems

[Bild von relationale Tabellenstruktur auf Google finden oder evntl. eigenes Projekt screenshotten]

## Abfrage-Engines

---

- Kombinierte Abfragesprachen
  - AQL
  - Gremlin
  - SQL
- ACID-Unterstützung

## Vorteile, Nachteile und Anwendungsfälle

---

### Vorteile

- Flexibel
- Reduzierte Komplexität
- Kostensparung
- Optimierte Integration
- Agilität
- Skalierbarkeit

### Nachteile

- Komplexe Architektur
- Leistungskompromisse
- Hohe Lernkurve
- Limitierte Features

### Anwendungsfälle

- Content-Management-Systeme
- Internet of Things
- Logistik und Lieferkettenmanagement
- Soziale Netzwerke
- Gesundheitswesen

## Verschiedene Systeme

---

- OrientDB
- Couchbase
- Microsoft Azure Cosmos DB
- ArangoDB

## OrientDB

- Open-Source
- Dokumente, Graphen und Objektorientierung
- SQL-ähnliche Abfragesprache

## Couchbase

- Key-Value- und dokumentenorientierte Speicherung
- N1QL als eigene Abfragesprache

## Microsoft Azure Cosmos DB

- Cloud-basiert, global verteilt
- Unterstützt verschiedene APIs und Modelle

## ArangoDB

- Ehemalig Open-Source
- Dokument-, Graph- und Key-Value-Modelle
- AQL als eigene Abfragesprache
- Wie man sieht, gibt es viele Systeme (auch CrateDB, EnterpriseDB, MarkLogic, SAP HANA, Virtuoso, SurrealDB und viele mehr)
- Unsere Wahl: ArangoDB

## ArangoDB

---

- Vor Oktober 2023 Open Source
- Gratis Community-Edition
- 2012 veröffentlicht
- Vereinigt drei Hauptmodelle: 1: Dokumentenorientiertes Modell (ähnlich wie MongoDB) 2: Graphorientiertes Modell 3: Key-Value-Modell
- Für klein und groß, skalierbar

## Warum ArangoDB

- Einheitliche Engine
- Einheitliche Abfragesprache (AQL)
- Einfache Installation & Cross-Plattform
- Skalierbarkeit

## Installation und erste Erfahrungen mit ArangoDB

---

### Installation

- Einfach über Installationspaket oder Docker
- Port 8529
- ArangoDB Web UI

## Besonderheiten

- Schnell wechselbar zwischen Graph- und Dokumentenansicht
- AQL-Abfragen sind bei bestehenden SQL-Grundkenntnissen einfach zum lernen
- Foxx Framework, für Microservices/REST-Services direkt in der Datenbank

## Einordnung im CAP-Dreieck

- Nicht direkt einordbar
- Anpassbare Konsistenz, flexible Verfügbarkeit
- Von Natur aus Partitionstolerant

# Konzepte und Schema von ArangoDB

---

## Dokumentenspeicherung

- JSON
- eindeutige `_key`-Eigenschaft
- optional `_id` bzw. `_ref`

## Graphenspeicherung

- Vertex-Collections (Knoten) und Edge-Collections (Kanten)
- Kanten verweisen auf `_from` und `_to`
- Mehrere Graphen können nebeneinander existieren

## Key-Value-Speicherung

- Dokumente auch als Key-Value-Paare behandelbar
- Super für teils unstrukturierte und schnelle Schlüssel-Wert-Zugriffe

# Schema-Definition von ArangoDB

---

## Schema-Definition

- Schemalos im Sinne von klassischen relationalen Datenbanken
  - Kein festes Schema erzwungen
  - Man kann "Flexibilität nutzen"
- Ab Version 3.7+: Möglichkeit auf Schema-Validierung

## Beziehungen

- Dokumentorientierte Beziehungen
  - Verweis auf `_id` eines anderen Dokuments

- Ähnlich wie Foreign-Keys
- Graphorientierte Beziehungen
  - Explizit in Edge-Collections gespeichert
  - Felder `_from` und `_to` -> Durchführung echter Graphabfragen (z.B. `GRAPH_TRAVERSAL`)

## Fazit

---

### Multi-Model Databases

- Ein nicht spezialisiertes System für mehrere Anwendungsfälle
- Flexibel, kostensparend, agil, skalierbar
- Komplex, Leistungsverluste, Lernkurve

### ArangoDB

- Starker Fokus auf Community und Entwicklung
- AQL als gemeinsame Abfragesprache
- Einfaches Setup, moderate Lernkurve
- Features wie Direkte Integration von Graphfunktionen, Web-UI, Foxx-Services für serverseitige JS-Lösungen, ...