



# ***Datenbanken*** ***NoSQL***

***S. Baldes***



# ***NoSQL-Datenbanken***

# Key-Value-DB

## Strategie: Key-Value

**Verfahren:** Zu einem Schlüssel wird ein Wert gespeichert.

**Einsatz:** Cachen aber auch als eigenständige persistente DB

**Vertreter:** Redis, Riak

Key ist meist ein String.

Value kann etwas Einfaches sein, aber auch ein komplizierte Objekt.

**Suche** nach Inhalten nicht möglich bzw. umständlich.

### Z.B.:

ANZAHLVERSUCHE	3
VERFAHRENSBESCHREIBUNG	c:\user\beschr.pdf
Kunde.1.telpriv	1234567

# Dokumentorientierte-DB

## Strategie: Dokumentenorientierte DB

**Verfahren:** Ein in beliebiger Syntax beschriebenes, schemafreies Datenkonstrukt

**Idee:** Zusammengehörige Daten bleiben zusammen (im Dokument)

**Einsatz:** Für Webdienste besonders geeignet.

**Vertreter:** MongoDB, CouchDB, Lotus Notes

Ein **Dokument** ist eine strukturierte Zusammenstellung bestimmter Daten.

Suche nach Inhalten möglich.

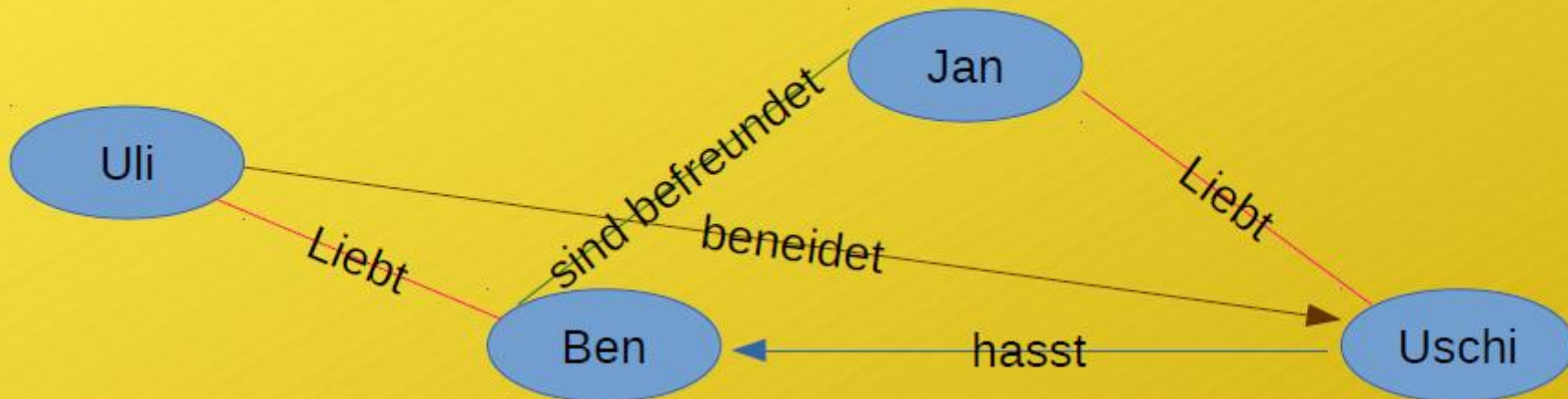
**Z.B.** (hier JSON -Format wie MongoDB):

```
{  
  „Vorname: „Uwe“,  
  „Nachname: „Müller“  
  „GebDat“: 13.3.2013  
}
```



# Graph-DB

## Strategie: Grafendatenbanken



**Verfahren:** DB zur Optimierung von Traversierungsproblemen.

**Idee:** Alternative für komplexe Abfragen in RDBMS mit i.d.R. mehrfach rekursiven Joins (teuer & kompliziert)

**Einsatz:** Hyper Link Strukturen, wer-kennt-wen, Navigation von Knoten zu Knoten

**Vertreter:** Neo4j, Titan u.v.a.



# mongoDB

**MongoDB** (abgeleitet vom engl.  
*humongous*, „gigantisch“ oder „riesig“)

# CRUD – Create

## CRUD – Create

### In der Shell

use WebNote

```
db.Notes.save(  
  {  
    Title: 'Mittag',  
    Message: 'nicht vergessen'  
  }  
);
```

# CRUD – Read

## CRUD – Read

Queries werden ebenso im Dokument-Stil spezifiziert

```
db.Notes.find();
```

```
db.Notes.find({ Title: /Test/ });
```

```
db.Notes.find(  
  { "Categories.Color": "red"}).limit(1);
```



# CRUD – Update

# CRUD – Update

```
> db.Notes.update({Title: 'Test'},
                  {'$set': {Categories: []}});
```

```
> db.Notes.update({Title: 'Test'},  
                  {'$push': {  
  
                      Categories:  
                        {Color: 'Red'}}  
                  },  
                  {});
```

# CRUD – Delete

## CRUD – Delete

```
db.dropDatabase();
```

```
db.Notes.drop();
```

```
db.Notes.remove({Title:"Mittag"});
```



# ***Anforderungen Prüfung***

# AP2 Wi2022/23 FA235 Aufgabe

## Analyse und Entwicklung von Netzwerken

- 3.1 Erläutern Sie die Unterschiede einer NoSQL-Datenbank gegenüber einer SQL-Datenbank hinsichtlich der Datenbanksprache, der Skalierbarkeit und der Datenstruktur.
- 3.2 Beschreiben Sie zwei Vorteile der Verwendung einer dokumentenorientierten Datenbank gegenüber einer SQL-Datenbank im Hinblick auf die geplante Anwendung.

# AP2 Wi2022/23 FA235 Lösung

## Analyse und Entwicklung von Netzwerken

### 3.1

	SQL-Datenbank	
Datenbanksprache	Hier wird SQL als weit verbreitete Sprache verwendet; SQL ist an Tabellenschema gebunden; SQL ist wenig flexibel;	Unterschiedliche zu SQL inkompatible Sprachen; größere Freiheiten in der Struktur; flexible Anpassung möglich; ...
Skalierbarkeit	Skalierung erfolgt meist durch Erweiterung des Speichers und Erhöhung der Rechenleistung (vertikale Skalierung)	Skalierung erfolgt i. d. R. horizontal, etwa durch Master-Slave-Struktur; Bildung von Clustern möglich;
Datenstruktur	Tabellenstruktur mit Beziehungen (Relationen) zueinander; Tabellen enthalten Attribute (Spalten) und Werte bzw. Datensätze (Zeilen)	Unterschiedliche Formen wie: Graphdatenbanken (z. B. Amazon Neptune) Dokumentenorientierte DB (z. B. MongoDB) Key-Value-DB (z. B. Google Big Table) etc.

### 3.2

Ein Vorteil der Dokumentenorientierten Datenbank gegenüber einer SQL-Datenbank liegt u. a. darin, dass neben den strukturierten Daten wie Temperaturverlauf auch unstrukturierte Daten wie Bilder, Filme, Streams flexibel abgelegt werden können. Ein zweiter Vorteil liegt in der besseren Skalierbarkeit der Dokumentenorientierten Datenbank. Da die Produktionsdaten sehr schnell anwachsen werden, ist abzusehen, dass eine Vergrößerung der Datenbank notwendig wird.

# AP2 Wi2022/23 FA235 Aufgabe Konzeption und Administration

- 3.1 Erklären Sie die Entwicklungsmodelle Agile Softwareentwicklung und das Wasserfallmodell.
- 3.2 Sie werden angehalten die Scrum-Methode auf Ihr Projekt anzuwenden.  
Erläutern Sie die einzelnen Rollen anhand passender Beispiele.



# AP2 Wi2022/23 FA235 Aufgabe

## Konzeption und Administration

- 3.1 Wasserfallmethode ist ein Vorgehensmodell im Projektmanagement. Das Projekt wird in unterschiedliche Phasen unterteilt und bereits zu Beginn bis ins Detail geplant. Dabei wird der Projektverlauf in verschiedene Phasen unterteilt, die inhaltlich aufeinander aufbauen. Agiles Projektmanagement eine Möglichkeit, Projekte zu planen und auf die Art wie Projekte abgewickelt werden müssen zur reagieren. In vielen Projekten gibt es Abweichungen vom Plan. Oft sind die Anforderungen an das Produkt (das Projektergebnis) zu Beginn des Projektes nicht vollends klar.
- 3.2 Product Owner
- Interessen der Anwender und Stakeholder (User Storys) genau zu kennen und konsequent zu vertreten.
  - Er betrachtet das Vorhaben strikt aus deren Perspektive.
  - Er ist Bindeglied zu Markt und Kunde.
  - Er entscheidet welche Produkteigenschaften bzw. -funktionalitäten entwickelt werden sollen und ob die jeweiligen Anforderungen vom Team vollständig erfüllt worden sind.
  - In den Entwicklungsprozess während der einzelnen Sprints indes greift er nicht ein.
- Scrum Master
- Fungiert als Moderator und Dienstleister und organisiert die Kommunikation des Entwicklungsteams mit der „Außenwelt“.
  - Er überwacht die Einhaltung der Werte und Regeln eines Projekts.
  - Er ist für geeignete Rahmenbedingungen zuständig.
- Entwicklungsteam
- Fünf bis zehn Projekt-Mitarbeitern.
  - Spezifische Hierarchien zwischen den einzelnen Kompetenzbereichen sind nicht vorgesehen.
  - Während der Sprints organisiert sich das Team selbst und realisiert eigenverantwortlich die jeweils geforderten neuen Produkt-Inkremente

# Quellen

- Lehrerfortbildung NoSQL, Manfred Steiner, Freiburg 2016
- Prüfungen BW AP2