

## C1G

Ich kann die Struktur von Daten in einer NoSQL Datenbank erläutern.

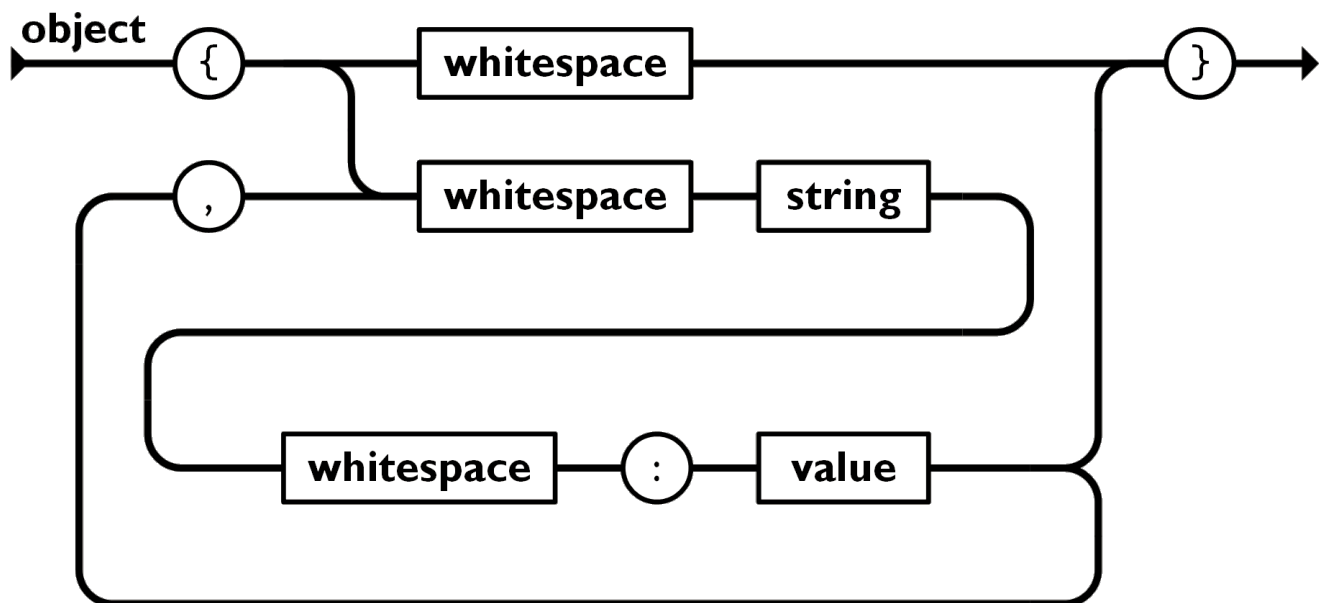
### Fragenstellung und Lernziele

- Was ist JSON und welche Rolle spielt es in NoSQL-Datenbanken?
- Wie ist die Struktur von JSON-Dokumenten aufgebaut?
- Wie werden Daten in einer dokumentenorientierten NoSQL-Datenbank gespeichert?
- Welche Unterschiede gibt es zwischen relationalen und NoSQL-Datenbanken?

### Umsetzung

NoSQL-Datenbanken wie MongoDB arbeiten mit einem flexiblen, schemalosen Datenmodell. Im Zentrum steht dabei das **JSON-Format**, das als Grundlage für die Darstellung und den Austausch von Daten dient.

### Grundlagen von JSON verstehen



**Definition und Bedeutung:** JSON (JavaScript Object Notation) ist ein leichtgewichtiges, textbasiertes Datenformat, das der Datenübertragung zwischen Anwendungen dient. Dieses Wissen ist grundlegend, um die Rolle von JSON in NoSQL-Datenbanken zu verstehen – ein zentrales Lernziel.

#### Syntax-Regeln:

- **Objekte** werden durch geschweifte Klammern `{ }` eingeschlossen.

- **Arrays** werden durch eckige Klammern `[]` dargestellt.
- **Key-Value-Paare** sind die Grundbausteine; der Schlüssel ist immer ein String und der Wert kann ein String, eine Zahl, ein Boolean, ein Array, ein Objekt oder `null` sein. Das detaillierte Verständnis dieser Regeln ermöglicht es, die Struktur von JSON-Dokumenten zu erklären.

#### Beispiel:

```
{
  "person": {
    "vorname": "Peter",
    "nachname": "Muster",
    "alter": 30,
    "hobbys": ["Lesen", "Reisen", "Programmieren"]
  }
}
```

## Gültige Elemente & Eigenschaften von Objekten und Arrays

**Gültige Elemente in JSON:** Ein JSON-Dokument darf nur bestimmte Datentypen enthalten, die als gültig definiert sind:

- Eine Zahl (integer oder floating point)
- Einen String (in doppelten Anführungszeichen)
- Einen Boolean (`true` oder `false`)
- Ein Array (in eckigen Klammern)
- Ein Objekt (in geschweiften Klammern)
- Den Wert `null`

Alle anderen Elemente, die nicht diesen Typen entsprechen, inklusive Kommentare mit `//`, gelten als ungültig. **Eigenschaften von Objekten:**

- Objekte werden durch geschweifte Klammern `{}` eingeschlossen.
- Sie bestehen aus Key-Value-Paaren, wobei der Schlüssel stets ein String sein muss.
- Mehrere Key-Value-Paare werden durch Kommas getrennt.
- Objekte können auch andere Objekte (verschachtelte Strukturen) als Werte enthalten.

#### Eigenschaften von Arrays:

- Arrays werden durch eckige Klammern `[]` eingeschlossen.
- Arrays können beliebige JSON-Werte enthalten, einschliesslich Objekte, Arrays, Zahlen, Strings, Booleans und `null`.
- Die Reihenfolge der Elemente in einem Array ist wichtig und bleibt erhalten.

## Struktur in einer NoSQL-Datenbank (z. B. MongoDB)

- **Dokumente und Collections:**
- **Dokumente:** Daten werden in JSON-ähnlichen Dokumenten gespeichert. Dies ermöglicht das flexible Speichern von Informationen
- **Collections:** Eine Collection ist eine Zusammenfassung von Dokumenten – vergleichbar mit Tabellen in relationalen Datenbanken, jedoch ohne festes Schema.
- **Beispiel eines MongoDB-Dokuments:**

```
{
  "_id": "unique_identifizier",
  "name": "Peter Muster",
  "kontakt": {
    "email": "peter.muster@example.com",
    "telefon": "0123456789"
  },
  "adresse": {
    "strasse": "Musterstrasse 12",
    "stadt": "Musterstadt",
    "plz": "12345"
  },
  "interessen": ["Lesen", "Reisen", "Programmieren"]
}
```

### Flexibilität der Struktur:

- **Individuelle Gestaltung:** Jedes Dokument kann unterschiedliche Felder enthalten, was die Anpassungsfähigkeit von NoSQL-Datenbanken unterstreicht.
- **Embedded Documents:** Informationen können innerhalb eines Dokuments verschachtelt werden, um komplexe Strukturen abzubilden.
- **Referenzen:** Statt Daten zu duplizieren, können Dokumente auch auf andere Dokumente verweisen.

## Vergleich zu relationalen Datenbanken

- **Schema und Struktur:** - Relationale Datenbanken verwenden ein fest definiertes Schema, was zu starren Datenstrukturen führt. - NoSQL-Datenbanken sind schemalos und erlauben variable Datenstrukturen.
- **Datenbeziehungen:** - Relationale Datenbanken nutzen **Joins**, um Beziehungen zwischen Tabellen herzustellen. - In NoSQL-Datenbanken erfolgt die Modellierung von Beziehungen häufig durch **Embedded Documents** oder **Referenzen**.
- **Speicherung:** - Relationale Datenbanken speichern Daten nicht im Klartext, sondern in optimierten binären Formaten, die für effiziente Abfragen und Speicherplatznutzung ausgelegt sind. - NoSQL-

Datenbanken verwenden ebenfalls binäre Formate, wie BSON (Binary JSON) in MongoDB, um die Daten effizient zu speichern und zu verarbeiten.

- **Skalierung:** - Relationale Systeme werden in der Regel vertikal skaliert (leistungsstärkere Server). - NoSQL-Datenbanken sind für horizontale Skalierung ausgelegt und können über mehrere Server verteilt werden.

## Wichtige Begriffe und Strukturen

- **JSON (JavaScript Object Notation):** Ein standardisiertes, textbasiertes Format zur Darstellung strukturierter Daten.
- **Dokumentenorientierte Datenbank:** Eine Datenbank, in der Daten als Dokumente (meist im JSON-Format) gespeichert werden.
- **Collection:** Eine Zusammenfassung von Dokumenten innerhalb einer Datenbank, vergleichbar mit einer Tabelle, jedoch ohne starres Schema.
- **Embedded Document:** Ein innerhalb eines Dokuments verschachteltes weiteres Dokument, das komplexe Strukturen erlaubt.
- **Referenz:** Ein Verweis von einem Dokument auf ein anderes, um Beziehungen zwischen Daten darzustellen.
- **Schema:** In relationalen Datenbanken fest definierte Strukturen; in NoSQL-Datenbanken fehlen diese Vorgaben, was zu erhöhter Flexibilität führt.