

Konzept zur Darstellung der Datenbank "Firma" in MongoDB

von Younes Hemeicheh (inf103656) und Marty Menge (stud106282)

1. Grundsätzliche Überlegungen

In MongoDB, einer dokumentenorientierten NoSQL-Datenbank, gelten andere Modellierungsprinzipien. Die wichtigsten Unterschiede und Überlegungen sind:

- **Normalisierung ist nicht zwingend erforderlich:** In relationalen Datenbanken werden Daten meist stark normalisiert, um Redundanz zu vermeiden und Konsistenz sicherzustellen. In MongoDB kann und sollte man hingegen denormalisieren, wenn es den Zugriff vereinfacht oder beschleunigt. Das bedeutet, zusammengehörige Informationen dürfen im selben Dokument gespeichert werden.
- **Fremdschlüssel sind optional:** MongoDB unterstützt keine Fremdschlüssel wie relationale Datenbanken. Stattdessen kann man Beziehungen über Referenzen (z. B. durch eine ID) oder durch eingebettete Dokumente modellieren. Die Wahl hängt davon ab, wie oft auf die verbundenen Daten zugegriffen wird und wie stark sie zusammengehören.
- **Abfrageorientiertes Schema:** Statt einer strikt logischen Modellierung steht in MongoDB das Abfrageverhalten im Vordergrund. Das Schema wird so gestaltet, dass typische Anfragen möglichst effizient ausgeführt werden können. Oft bedeutet das: Einbettung statt Trennung, Redundanz statt Normalisierung.

2. Modellierung der Firma-Datenbank in MongoDB

Basierend auf dem ER-Modell und den typischen Anwendungsfällen empfiehlt sich ein **Hybridmodell**:

Haupt-Collection: `personal`

Diese Collection enthält alle Kerndaten über Mitarbeiter und eingebettete Subdokumente:

```
{
  "pnr": "P123",
  "name": "Müller",
  "vorname": "Anna",
  "krankenkasse": "TK",
  "abteilung": {
    "abt_nr": "A01",
    "name": "Vertrieb"
  },
  "gehalt": {
    "gehaltsstufe": "E3",
    "betrag": 4200
  }
}
```

```

},
"kinder": [
  { "name": "Lukas", "vorname": "Müller", "geburtstag": "2012-05-01" }
],
"praemien": [
  { "pnr": "PR1", "betrag": 500 }
]
}

```

Zusätzliche Collection: maschinen

Diese Collection wird separat gehalten, da Maschinen unabhängig gepflegt und eventuell von mehreren Mitarbeitern genutzt oder zugeordnet werden könnten:

```

{
  "mnr": "M100",
  "name": "Bohrmaschine X",
  "neuwert": 10000,
  "zeitwert": 7000,
  "ansch_datum": "2021-01-01",
  "verantwortlich_pnr": "P123"
}

```

3. Bewertung

Kriterium	Hybridmodell
Performance (Lesen)	Hoch: eingebettete Daten vermeiden Joins
Performance (Schreiben)	Gut: Stammdaten selten geändert, zentral gepflegt
Wartbarkeit	Gut: klare Trennung zwischen stabilen Referenzen und dynamischen Daten
Konsistenz	Hoch: wenig Redundanz bei Gehalt, Abteilung

4. Fazit

Die Firma-Datenbank muss in MongoDB **nicht strikt in Normalform** vorliegen. Beziehungen müssen nicht zwingend über klassische Fremdschlüssel modelliert werden. Stattdessen ist eine **abfrageorientierte Struktur** sinnvoll. Das gewählte Hybridmodell erlaubt effiziente Lesezugriffe, gute Wartbarkeit und nutzt die Stärken von MongoDB optimal aus.