

## B1F

Ich kann ein vorgegebenes Datenmodell mit einer NoSQL Datenbank umsetzen.

### Fragenstellung und Lernziele

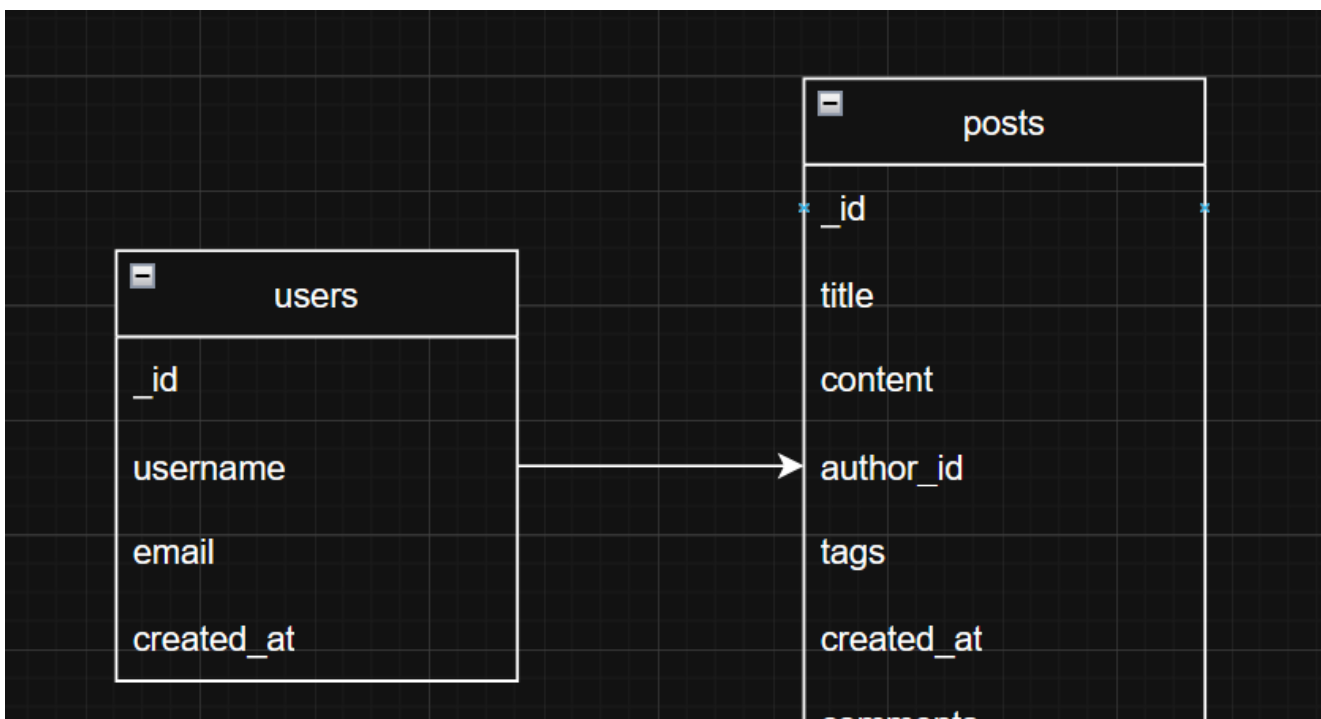
- Wie kann ich ein vorgegebenes Datenmodell in einer NoSQL-Datenbank umsetzen? **Ich kann ein vorgegebenes Datenmodell in einer NoSQL-Datenbank umsetzen.**
- Wie erstelle ich eine NoSQL-Datenbank? **Ich kann eine NoSQL-Datenbank erstellen.**
- Wie speichere ich Daten in einer NoSQL-Datenbank? **Ich kann Daten in einer NoSQL-Datenbank speichern.**

### Umsetzung

Um ein vorgegebenes Datenmodell in einer NoSQL-Datenbank umzusetzen, sind folgende Schritte erforderlich:

- Die Struktur des Datenmodells verstehen und analysieren.
- Die NoSQL-Datenbank auswählen, die am besten zu den Anforderungen des Projekts passt. (MongoDB)
- Eine NoSQL-Datenbank erstellen und konfigurieren.

Das Datenmodell sieht wie folgt aus:



comments

### + users



```
{
  "_id": ObjectId,
  "username": "johndoe",
  "email": "john@example.com",
  "created_at": ISODate
}
```

### + posts



```
{
  "_id": ObjectId,
  "title": "My First Post",
  "content": "This is the content of the post.",
  "author_id": ObjectId,
  "tags": ["mongodb", "nosql"],
  "created_at": ISODate,
  "comments": [
    {
      "author_id": ObjectId,
      "content": "Nice post!",
      "created_at": ISODate
    },
    {
      "author_id": ObjectId,
      "content": "Thanks for sharing!",
      "created_at": ISODate
    }
  ]
}
```

Um dieses Datenmodell in einer NoSQL-Datenbank umzusetzen, muss eine Datenbank erstellt werden.

Um eine MongoDB-Datenbank zu erstellen kann ein Docker-Container verwendet werden:

### + docker-compose.yml



```
services:
```

```
mongo:
  image: mongo:7
  container_name: mongoddb
  ports:
    - "27017:27017"
  volumes:
    - mongo-data:/data/db

volumes:
  mongo-data:
```

Docker Desktop erlaubt es, ein Terminal zu öffnen, um mit der MongoDB-Datenbank zu kommunizieren. Hier können die Daten in die Datenbank eingefügt werden.

+ **bash**



```
# connect to the MongoDB container
mongosh

# switch to the database
use blogdb

# insert a user
const userId = ObjectId();
db.users.insertOne({
  _id: userId,
  username: "johndoe",
  email: "john@example.com",
  created_at: new Date()
});

# insert a post
db.posts.insertOne({
  title: "My First Post",
  content: "This is the content of the post.",
  author_id: userId,
  tags: ["mongodb", "nosql"],
  created_at: new Date(),
  comments: [
    {
      author_id: userId,
      content: "Nice post!",
      created_at: new Date()
    }
  ]
});
```

Somit sind die Daten in der MongoDB-Datenbank gespeichert. Die Daten können nun abgerufen und bearbeitet werden.

```
blogdb> db.posts.findOne({ title: "My First Post" })
...
{
  _id: ObjectId('681c94b84a34fa337dd861e1'),
  title: 'My First Post',
  content: 'This is the content of the post.',
  author_id: ObjectId('681c94ac4a34fa337dd861e0'),
  tags: [ 'mongodb', 'nosql' ],
  created_at: ISODate('2025-05-08T11:25:44.546Z'),
  comments: [
    {
      author_id: ObjectId('681c94ac4a34fa337dd861e0'),
      content: 'Nice post!',
      created_at: ISODate('2025-05-08T11:25:44.546Z')
    }
  ]
}
blogdb>
```

## Zusammenfassung

NoSQL-Datenbanken wie MongoDB verwenden eine flexible, schemalose Struktur, die es ermöglicht, Daten in Form von Dokumenten zu speichern. Diese Dokumente können komplexe Strukturen enthalten, wie z. B. eingebettete Dokumente und Arrays.

Ein Datenmodell umzusetzen ist daher ein einfacher Prozess, der es ermöglicht, Daten in einer NoSQL-Datenbank zu speichern und abzurufen. Die Flexibilität und Skalierbarkeit von NoSQL-Datenbanken machen sie zu einer idealen Wahl für moderne Anwendungen.