

## C1F

Ich kann Daten in eine NoSQL Datenbank übernehmen.

### Fragenstellung und Lernziele

- **Welche Methoden gibt es, um Daten in eine NoSQL Datenbank zu importieren?** Ich kann verschiedene Ansätze zum Import von Daten in NoSQL Datenbanken nennen.
- **Wie wird der Datenimport praktisch umgesetzt?** Ich kann den Einsatz von Tools wie `mongoimport`, Mongo Express oder Mongo Compass erläutern.
- **Welche Datenformate werden für den Import unterstützt?** Ich kann die häufig genutzten Formate nennen und ihre Besonderheiten erläutern.
- **Wie kann man den Erfolg des Datenimports validieren?** Ich kann erklären, wie durch Abfragen und Überprüfungen sichergestellt wird, dass die importierten Daten korrekt und vollständig übernommen wurden.

### Umsetzung

In NoSQL Datenbanken erfolgt die Datenübernahme über verschiedene Methoden. Häufige Ansätze umfassen:

- **CLI-Tools:** Der Einsatz von Befehlszeilentools wie `mongoimport` ermöglicht einen automatisierten und skriptbaren Datenimport.
- **GUI-basierte Tools:** Anwendungen wie **Mongo Express** oder **Mongo Compass** bieten eine benutzerfreundliche Oberfläche, um Dateien (z. B. JSON oder CSV) in eine Datenbank zu importieren.
- **Programmgesteuerten Import:** Durch Skripte in Programmiersprachen wie Python oder JavaScript können Daten programmatisch in die Datenbank übernommen werden.

### Beispiel: Import einer JSON-Datei mit mongoimport

#### 1. Vorbereitung:

Stellen Sie sicher, dass die JSON-Datei korrekt formatiert ist und alle erforderlichen Daten enthält.

Beispiel für eine JSON-Datei (`data.json`):

```
{
  "name": "Max Mustermann",
  "email": "max.mustermann@example.com",
  "age": 30
}
```

```
    age: 30,  
    "hobbies": ["Lesen", "Reisen", "Programmieren"]  
  },  
  {  
    "name": "Anna Musterfrau",  
    "email": "anna.musterfrau@example.com",  
    "age": 25,  
    "hobbies": ["Fotografie", "Kochen", "Wandern"]  
  }  
]
```

## 2. Ausführung:

Führen Sie folgenden Befehl aus, um die Daten in die gewünschte Datenbank und Collection zu übernehmen:

```
mongoimport --db exampleDB --collection documents --file data.json --json
```

## Validierung des Datenimports

- **Validierung:** Prüfen Sie mittels Abfragen, ob die importierten Dokumente vollständig und korrekt in der Collection vorhanden sind.

Nach dem Import der Daten in die NoSQL-Datenbank kann die folgende Abfrage verwendet werden, um sicherzustellen, dass die Daten korrekt übernommen wurden:

```
// Überprüfen, ob die Collection Dokumente enthält  
db.documents.find().pretty();
```

### Erklärung:

`db.documents.find()` ruft alle Dokumente aus der Collection `documents` ab.

`.pretty()` formatiert die Ausgabe, um sie lesbarer zu machen.

**Zusätzliche Validierung:** Falls spezifische Kriterien überprüft werden sollen, kann eine gezielte Abfrage verwendet werden. Zum Beispiel:

```
// Überprüfen, ob ein bestimmtes Dokument importiert wurde  
db.documents.find({ "name": "Max Mustermann" });
```

### Erwartetes Ergebnis:

- Die Abfrage sollte das Dokument mit dem Namen `Max Mustermann` zurückgeben, falls der Import erfolgreich war.
- Falls keine Ergebnisse zurückgegeben werden, sollte der Importprozess erneut überprüft werden.

### Hinweis:

Falls die Daten nicht wie erwartet erscheinen, können folgende Schritte helfen:

1. Überprüfen Sie den Importbefehl und die verwendete Datei.
2. Stellen Sie sicher, dass die Datenbank und die Collection korrekt benannt sind.
3. Prüfen Sie die Logs auf mögliche Fehler während des Imports.

## Wichtige Begriffe und Strukturen

- **Datenimport:** Der Prozess der Übernahme von Daten aus externen Quellen (z. B. Dateien) in eine NoSQL Datenbank.
- **CLI-Tools:** Befehlszeilenprogramme wie `mongoimport`, die den automatisierten Import von Daten ermöglichen.
- **GUI-basierte Tools:** Anwendungen mit grafischer Oberfläche, wie Mongo Express oder Mongo Compass, die den Import von Daten vereinfachen.
- **Validierung:** Methoden zur Überprüfung, ob die importierten Daten korrekt und vollständig übernommen wurden.
- **Fehlerbehandlung:** Mechanismen, um während des Importprozesses auftretende Probleme zu identifizieren und zu beheben.

Der Import von Daten in eine NoSQL Datenbank ist ein zentraler Prozess, der über unterschiedliche Methoden erfolgen kann. Ob über CLI-Tools, grafische Anwendungen oder programmatische Ansätze – durch den Einsatz standardisierter Datenformate wie JSON oder CSV und einer anschließenden Validierung kann sichergestellt werden, dass die Datenübernahme effizient und fehlerfrei erfolgt.

## Nachweis

1. **Praktische Übung:** Importiere eine JSON- oder CSV-Datei in eine NoSQL Datenbank (z. B. MongoDB) mittels `mongoimport` oder einem GUI-Tool.

## Testdb

```
version: '3.8'

services:
  mongodb:
    image: mongo:7
    container_name: test_mongo
    ports:
      - "27017:27017"
    volumes:
      - mongo_data:/data/db
      - ./example.csv:/tmp/example.csv:ro
    restart: unless-stopped
```

```
volumes:
  mongo_data:
```

```
docker-compose up -d
```

```
docker exec test_mongo mongoimport `
  --db testdb `
  --collection people `
  --type csv `
  --headerline `
  --file /tmp/example.csv
```

## Sample CSV

```
name,age,city
Alice,30,New York
Bob,25,San Francisco
Charlie,35,Berlin
```

```
docker exec -it test_mongo mongosh
# in der MongoDB Shell
use testdb
db.people.find().pretty()
```

```
nTimeoutMS=2000&appName=mongosh+2.5.0
```

```
Using MongoDB:      8.0.6
```

```
Using Mongosh:      2.5.0
```

For mongosh info see: <https://www.mongodb.com/docs/mongodb-shell/>

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (<https://www.mongodb.com/legal/privacy-policy>).  
You can opt-out by running the `disableTelemetry()` command.

---

### The server generated these startup warnings when booting

```
2025-04-10T12:00:36.954+00:00: Using the XFS filesystem is strongly recommended with
the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2025-04-10T12:00:37.709+00:00: Access control is not enabled for the database. Read
and write access to data and configuration is unrestricted
2025-04-10T12:00:37.709+00:00: For customers running the current memory allocator, w
e suggest changing the contents of the following sysfsFile
2025-04-10T12:00:37.709+00:00: We suggest setting the contents of sysfsFile to 0.
2025-04-10T12:00:37.709+00:00: vm.max_map_count is too low
2025-04-10T12:00:37.709+00:00: We suggest setting swappiness to 0 or 1, as swapping
can cause performance problems.
```

---

```
test> use testdb
... db.people.find().pretty()
...
switched to db testdb
testdb> db.people.find().pretty()
[
  {
    _id: ObjectId('67f7b31b69a415fa2f96de0c'),
    name: 'Alice',
    age: 30,
    city: 'New York'
  },
  {
    _id: ObjectId('67f7b31b69a415fa2f96de0d'),
    name: 'Bob',
    age: 25,
    city: 'San Francisco'
  },
  {
    _id: ObjectId('67f7b31b69a415fa2f96de0e'),
    name: 'Charlie',
    age: 35,
    city: 'Berlin'
  }
]
```