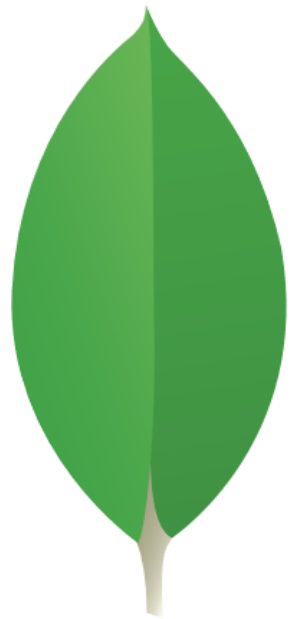


Grundlagen



mongoDB

Lernziele



- Aufbau einer MongoDB – als Beispiel einer dokumentenbasierten DB
- Installation MongoDB
- Server Starten
- Die Mongo Shell
- CRUD Funktionen

Grundlagen



Datenbanken

- Ein MongoDB-Server verwaltet mehrere logische Datenbanken.

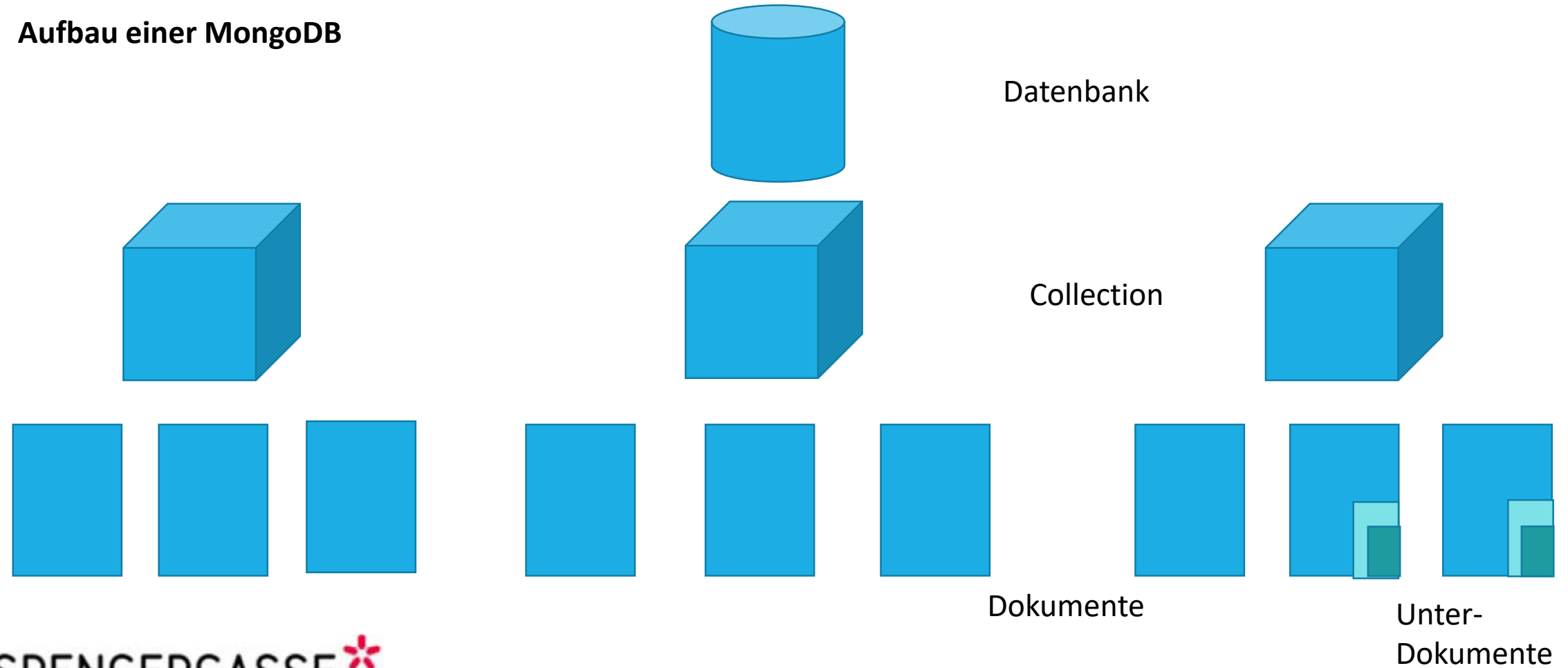
Collection

- Einzelne Dokumente werden innerhalb logischer **Namensräume** organisiert = **Collection**
- Entspricht zwar einer Tabelle einer relationalen DB ist aber nicht identisch
- Collections müssen nicht explizit angelegt werden – sie werden bei der ersten Verwendung automatisch erzeugt.

Grundlagen



Aufbau einer MongoDB



Grundlagen



Dokumente

- MongoDB = Dokumentenorientiert
- Dokument entspricht einem Datensatz
- Speichert Dokumente intern in BSON ab. (BSON = Erweiterung von JSON – Serialisierungsformat in JavaScript)
 - `> doc = {hello: „MongoDB“}`
 - Doc = Dokument
 - Hello = Feld
 - „MongoDB“ = String-Wert des Feldes „Hello“

Installation



1. Aktuelle Version herunterladen

1. <http://www.mongodb.org/downloads>
2. Wählen Sie MongoDB Community Server
3. Entsprechendes Betriebssystem auswählen
4. **MSI** Paket wählen

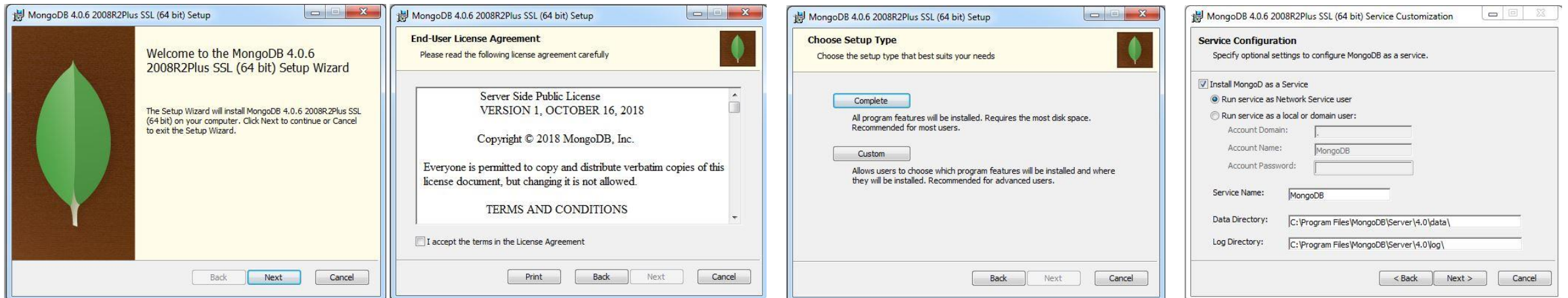
2. Inhalt der Datei in ein frei wählbares Verzeichnis entpacken

1. Z.B. C:\Programme\MongoDB
2. Es wird ein neues Verzeichnis mit dem Namen „mongodb“ gefolgt von Betriebssystem- und Versionsinfos erstellt.

Installation



- Den Anweisungen des Installations Wizards folgen



Achtung: Checkbox „Mongo Compass installieren“ DEAKTIVIEREN!

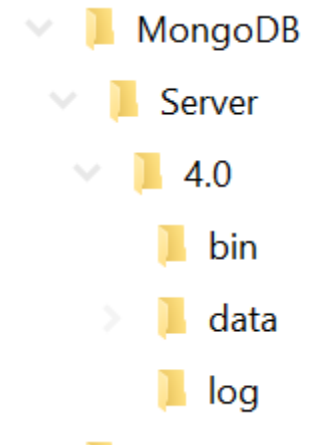
Starten



Verzeichnis Data anlegen (im Root Verzeichnis)

Server starten

- Eingabeaufforderung starten
- In jenes Verzeichnis wechseln in dem Sie MongoDB installiert haben
„`cd mongodb\server\4.2\bin`“
- Befehl: „`mongod --dbpath \data`“ eingeben
(Achtung das directory „data“ ist hier im root Verzeichnis)

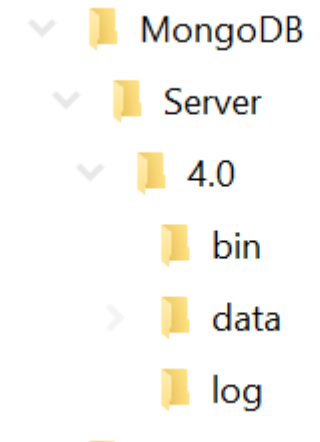


Starten



Mongo Shell starten

- Eingabeaufforderung
- In jenes Verzeichnis wechseln in dem Sie MongoDB installiert haben
„`cd mongodb\server\4.2\bin`“
- Befehl: „`mongo`“ eingeben



Erste Schritte



- Im MongoDB Server werden mehrere Datenbanken verwaltet.
- Befehl: „**show dbs**“
 - Solange wir noch keine neue DB angelegt haben wird die stets existierende Datenbank „local“ angezeigt
- Befehl: „**help**“
 - Gibt eine Übersicht über weitere Befehle
- Befehl: „**show collections**“
 - Zeigt alle verfügbaren Collections.
 - Diese müssen nicht explizit angelegt werden → werden automatisch erzeugt .

Erste Schritte – CRUD-Operationen



Insert

- Befehl:
 - `db.dokumente.insert ({a:1, b:"zwei"})`
 - `db.dokumente.insert ({a:2, b:"drei"})`
- **db** = aktuell selektierte Datenbank
- **collections** werden automatisch zu den Attributen dieses Datenbankobjektes
- → Syntax: `db.name`
- Mit dem ersten schreibenden Zugriff auf die Collection **dokumente** wurde diese angelegt

Hinweis: Wenn Sie keine Datenbank ausgewählt haben arbeiten Sie automatisch in der Test Datenbank.

Erste Schritte – CRUD-Operationen



Zählen:

- Befehl: `db.dokumente.count()`

Suchen:

- Befehl: `db.dokumente.find()`
 - Was fällt Ihnen auf, wenn Sie diesen Befehl ausführen?

```
> db.dokumente.find()
{ "_id" : ObjectId("5c6da62cc19c828b8d4f1e3d"), "a" : 1, "b" : "zwei" }
{ "_id" : ObjectId("5c6da6f0c19c828b8d4f1e3e"), "a" : 2, "b" : "drei" }
>
```

- Beim Schreiben wurde das Feld „_id“ erzeugt → dient zur eindeutigen Identifizierung

Erste Schritte – CRUD-Operationen



Suchen:

Man kann der Methode „find(...)“ Suchkriterien und Feldgruppen in Form von Dokumenten mitgeben, um die Treffermenge einzuschränken

- Befehl: `db.dokumente.find({a:2}, {_id:0})`

- Die Suche wird auf die Dokumente eingeschränkt, deren Feld „a“ den Wert 2 haben.

```
> db.dokumente.find({a:2}, {_id:0})
{ "a" : 2, "b" : "drei" }
```

- Mit dem optionalen zweiten Parameter „_id:0“ wird die Suche innerhalb eines Dokumentes eingeschränkt, indem für Felder, die ignoriert werden sollen eine 0 angegeben wird.
 - **Ergebnis ohne zweiten Parameter:**

```
> db.dokumente.find({a:2})
{ "_id" : ObjectId("5c6da6f0c19c828b8d4f1e3e"), "a" : 2, "b" : "drei" }
>
```

Erste Schritte – CRUD-Operationen



Arrays / eingebettete Unterobjekte:

Im Gegensatz zu relationalen Datenbanken kann ein einzelnes Feld eines Dokuments auch Arrays oder eingebettete Unterobjekte enthalten.

```
db.autoren.insert({  
  titel: „Der Name der Rose“,  
  Autoren :[  
    {name: „N.N.“},  
    {name: „Umberto Ecco“}  
  ]  
})
```

Erste Schritte – CRUD-Operationen



Suche mit formatierter Ausgabe:

- Befehl: `db.autoren.find().pretty()`

```
> db.autoren.find().pretty()
{
  "_id" : ObjectId("5c6dcaf0c19c828b8d4f1e3f"),
  "titel" : "Der Name der Rose",
  "Autoren" : [
    {
      "name" : "N.N"
    },
    {
      "name" : "Umberto Ecco"
    }
  ]
}

{
  "_id" : ObjectId("5c6eab7e2f2f2eb9ae65806c"),
  "titel" : "Biologie einer Begegnung",
  "Autoren" : [
    {
      "name" : "N.N"
    },
    {
      "name" : "Axel Barth"
    }
  ]
}
```

Erste Schritte – CRUD-Operationen



Update:

Zunächst wollen wir uns die Collection „dokumente“ so anschauen, wie wir sie erstellt haben. Führen Sie dazu eine uneingeschränkte Suche auf die Collection „dokumente“ aus.

- Befehl: `db.dokumente.find()`

```
> db.dokumente.find()
{ "_id" : ObjectId("5c6da62cc19c828b8d4f1e3d"), "a" : 1, "b" : "zwei" }
{ "_id" : ObjectId("5c6da6f0c19c828b8d4f1e3e"), "a" : 2, "b" : "drei" }
```

Erste Schritte – CRUD-Operationen



Update:

```
> db.dokumente.find()
{ "_id" : ObjectId("5c6da62cc19c828b8d4f1e3d"), "a" : 1, "b" : "zwei" }
{ "_id" : ObjectId("5c6da6f0c19c828b8d4f1e3e"), "a" : 2, "b" : "drei" }
```

Im nächsten Schritt wollen wir ein bestimmtes Dokument verändern. Führen Sie den folgenden Update Befehl aus und danach noch einmal die uneingeschränkte Suche.

- Befehl: `db.dokumente.update({a:2}, {b:"vier"})`

```
> db.dokumente.find()
{ "_id" : ObjectId("5c6da62cc19c828b8d4f1e3d"), "a" : 1, "b" : "zwei" }
{ "_id" : ObjectId("5c6da6f0c19c828b8d4f1e3e"), "a" : 2, "b" : "vier" }
```

Achtung:

Die Semantik der `update(...)` – Methode ist so, dass der zweite Parameter (hier `{b:"vier"}`) das bestehende Dokument **vollständig ersetzt**

Erste Schritte – CRUD-Operationen



Update Partiiell:

Im nächsten Schritt wollen wir in einem bestimmten Dokument ein Attributsinhalt verändern. Führen Sie den folgenden Update Befehl aus und danach noch einmal die uneingeschränkte Suche.

- Befehl: `db.dokumente.update({a:1}, {$set: {b:"fünf"}})`

```
> db.dokumente.find()
{ "_id" : ObjectId("5cde9fd1ca8b55448eee1767"), "a" : 1, "b" : "fünf" }
{ "_id" : ObjectId("5cdea017ca8b55448eee1768"), "b" : "vier" }
```

Aufgabe: Versuchen Sie das Dokument: a:1, b: „fünf“ auf a:4, b: „sechs“ zu ändern

- Befehl: `db.dokumente.update({a:1}, {$set: {a:4, b:"sechs"}})`

```
{ "_id" : ObjectId("5cde9fd1ca8b55448eee1767"), "a" : 4, "b" : "sechs" }
{ "_id" : ObjectId("5cdea017ca8b55448eee1768"), "b" : "vier" }
```

Erste Schritte – CRUD-Operationen



Aufgabe:

Lassen Sie sich alle Dokumente ohne ID anzeigen

```
> db.dokumente.find()
{ "_id" : ObjectId("5c6da62cc19c828b8d4f1e3d"), "a" : 1, "b" : "zwei" }
{ "_id" : ObjectId("5c6da6f0c19c828b8d4f1e3e"), "b" : "vier" }
```

Sie erinnern sich? Wir geben den Parameter `{}` an, um die Menge der zurückgegebenen Dokumente zu steuern. Hier also: `{}`

Finde alle Dokumente

Diese Felder nicht anzeigen

- Befehl: `db.dokumente.find({}, {_id:0})`

```
> db.dokumente.find({}, {_id:0})
{ "a" : 1, "b" : "zwei" }
{ "b" : "vier" }
```

Erste Schritte – CRUD-Operationen



Löschen:

Einzelne Dokumente werden mit der Methode: „remove(...)“ gelöscht:

Befehl: `db.dokumente.remove ({a:4})`

Kontrollieren Sie das Ergebnis mit der „find“ Methode

Befehl: `db.dokumente.find({}, {_id:0})`

```
> db.dokumente.remove({a:1})
WriteResult({ "nRemoved" : 1 })
> db.dokumente.find({}, {_id:0})
{ "b" : "vier" }
```

Erste Schritte – CRUD-Operationen



Löschen:

Alle Dokumente einer Collection können mit der Methode: „remove()“ - ohne Parameter – gelöscht werden. Das kann, aufgrund der Datenmenge, recht lange dauern da jeder Datensatz einzeln gelöscht wird und auch die Aktualisierung der Indizes zur Folge haben kann...

Schneller geht es mit der Methode „drop“. Hierbei werden alle Dokumente inklusive bestehender Indizes gelöscht.

Befehl: `db.dokumente.drop ()`

Kontrollieren Sie das Ergebnis mit dem

Befehl: `show collections`