



Von: Matthias Hrbek

# Agenda

- Was ist MongoDB?
- Features von MongoDB
- Begriffe
- MongoDB Datentypen
- Wer verwendet MongoDB?
- Queries
- Aggregation
- MongoDB Compass & Atlas



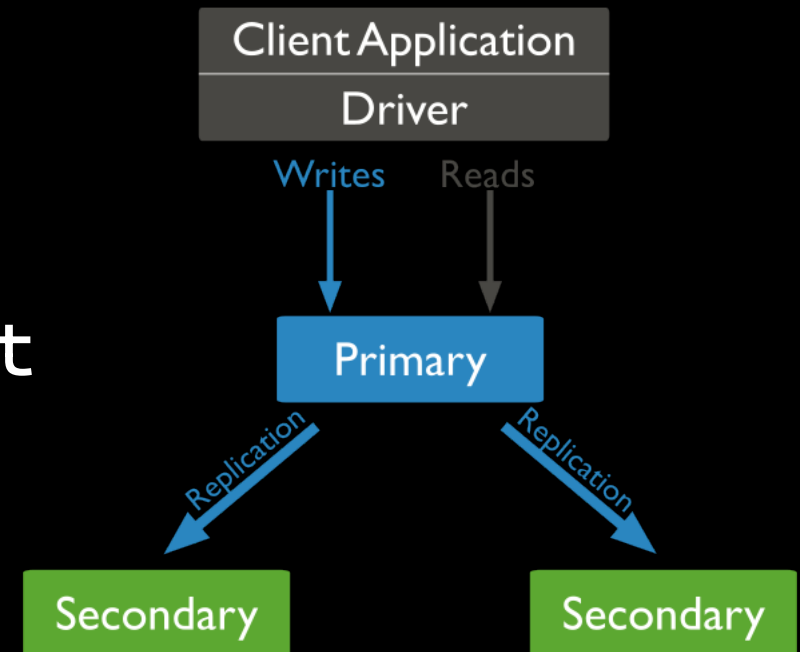
# Was ist MongoDB überhaupt?

- Bekanntestes NoSQL Datenbanksystem
- Open-source
- Document based
- BSON anstatt von JSON
- Sehr cool



# Features von MongoDB

- Indexing
- Skalierbarkeit
- Replikation & Hohe Verfügbarkeit
- Lastverteilung
- Aggregation
- Breiter Support an Programmiersprachen



# Wer verwendet MongoDB?

MongoDB wird als Backend-Datenbank von vielen globalen Unternehmen betrieben.

- Toyota
- The New York Times
- SourceForge
- Etsy
- CERN
- Vodafone
- Coinbase
- ...



# MongoDB Datentypen

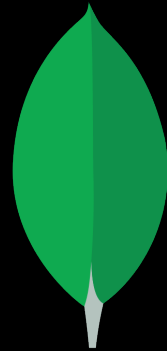
- Date
- ObjectID
- Double

- Int32
- Long
- Decimal128
- Timestamp



# Begriffe

- Databases
- Collections
- Documents
- Field Names



mongoDB®

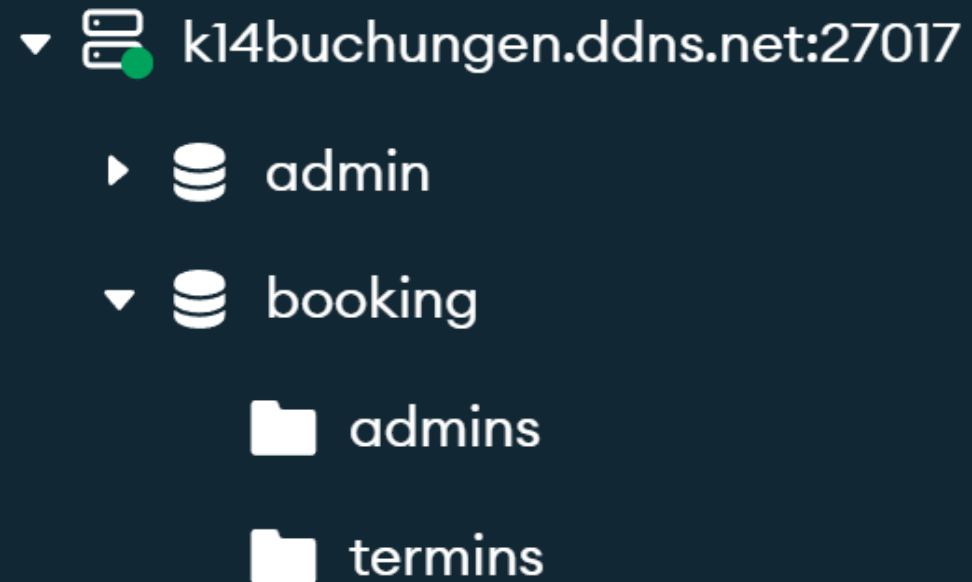




# Beispiel: Begriffe

Server → Database → Collection

Document + Fields



```
personen: [  
  {  
    vorname: 'Matthias',  
    nachname: 'Hrbek',  
    tel: '06702007320',  
    _id: ObjectId('67ba1dba028d58cfbcfde4c0')  
  }  
],
```



# SQL X MongoDB Begriffe

| SQL         | MongoDB               |
|-------------|-----------------------|
| Database    | Database              |
| Table       | Collection            |
| Index       | Index                 |
| Row         | BSON Document         |
| Column      | BSON Field            |
| Primary Key | _id field             |
| Group by    | Aggregation           |
| Join        | Embedding and Linking |



# Queries

# List

## Datenbanken auflisten – `show dbs`

Zeigt alle Datenbanken, die auf dem verbundenen Datenbankserver abgespeichert sind.

## Collections auflisten – `show collections`

Datenbank muss vorher ausgewählt worden sein.  
Zeigt alle verfügbaren Collections einer Datenbank.

## Documents auflisten – `db.collection.find()`

Zeigt alle Datensätze einer Collection an.

```
> show dbs
```

```
< admin      168.00 KiB
   booking   132.00 KiB
   cars       40.00 KiB
   config     72.00 KiB
   local      88.00 KiB
```

```
> show collections
```

```
< models
```

```
> db.models.find()
< {
  _id: ObjectId('67ba37a80f9927330f58aa49'),
  brand: 'Tesla',
  model: 'Model S',
  year: 2022,
  price: 89999.99,
  engine: {
    type: 'Electric',
    power: 670,
    cylinders: 0
  }
}
```

# Create

## Datenbank erstellen – `use [name]`

Wenn eine Datenbank mit dem angegebenen Namen existiert wird diese ausgewählt, falls nicht, wird diese erstellt.

## Collection erstellen – `db.createCollection("name")`

Kann, aber muss nicht verwendet werden um Collections zu erstellen.

`db.name.insert(object)` kann ebenso zum erstellen und simultanem befüllen verwendet werden.

```
> use animals  
< switched to db animals
```

```
> db.createCollection("mammoths")  
< { ok: 1 }
```

# Drop

## Datenbank löschen – `db.dropDatabase()`

Löscht die aktuell ausgewählte Datenbank und gibt `{ "ok": 1 }` zurück.

## Collection löschen – `db.collection.drop()`

Löscht die angegebene Collection und gibt bei Erfolg `true` zurück.

```
> db.dropDatabase()  
< { ok: 1, dropped: 'animals' }
```

```
> db.mammoths.drop()  
< true
```

# Create Documents

## Dokument erstellen – `db.collection.insertOne({})`

Fügt ein Dokument in die ausgewählte Collection ein. Gibt `{ acknowledged: true }` und die automatisch generierte ObjectID zurück.

## Mehrere erstellen – `db.collection.insertMany([{}, {}])`

Fügt mehrere Dokumente in die ausgewählte Collection ein. Gibt ebenso ein `{ acknowledged: true }` und die automatisch generierten ObjectIDs zurück.

```
animals> db.mammoths.insertOne({
  "species": "Mammuthus primigenius",
  "commonName": "Wolliges Mammut",
  "habitat": ["Tundra", "Steppe"],
  "height_m": 4.0,
  "weight_kg": 6000
})
```

```
< {
  acknowledged: true,
  insertedId: ObjectId('67bb49192b05426a9fd06ca9')
}
```

# Datensätze lesen (.find())

Alle Dokumente anzeigen – `db.collection.find()`

Formatiert – `db.collection.find().pretty()`

Gefiltert – `db.collection.find({ "height_m": 4.0 })`

Ausgabe gefiltert –

```
db.models.find({"sold": false}, {"brand": 1, "model": 1})
```



# Filteroptionen

| Operator           | Bedeutung                   | Beispiel  |
|--------------------|-----------------------------|---|
| <code>\$eq</code>  | Gleich                      | <code>{ brand: { \$eq: „Ford“ } }</code>            |
| <code>\$ne</code>  | Ungleich                    | <code>{ brand: { \$ne: "BMW" } }</code>             |
| <code>\$gt</code>  | Größer als                  | <code>{ year: { \$gt: 2020 } }</code>               |
| <code>\$gte</code> | Größer/gleich               | <code>{ year: { \$gte: 2020 } }</code>              |
| <code>\$lt</code>  | Kleiner als                 | <code>{ price: { \$lt: 50000 } }</code>             |
| <code>\$lte</code> | Kleiner/gleich              | <code>{ price: { \$lte: 50000 } }</code>            |
| <code>\$in</code>  | Werte in Liste<br>enthalten | <code>{ brand: { \$in: ["Tesla", "Ford"] } }</code> |
| <code>\$nin</code> | Werte nicht in Liste        | <code>{ brand: { \$nin: ["BMW", "Audi"] } }</code>  |

# Logikoperatoren

| Operator           | Bedeutung | Beispiel  |
|--------------------|-----------|---|
| <code>\$and</code> | Und       | <code>{ \$and: [{ year: { \$gte: 2020 } }, { price: { \$lt: 60000 } }] }</code> |
| <code>\$or</code>  | Oder      | <code>{ \$or: [{ brand: "Tesla" }, { brand: "BMW" }] }</code>                   |
| <code>\$not</code> | Negation  | <code>{ price: { \$not: { \$gt: 50000 } } }</code>                              |

# Dokumente aktualisieren

## Grundmethoden für Updates:

`updateOne(filter, update)`

Aktualisiert das erste passende Dokument

`updateMany(filter, update)`

Aktualisiert alle passenden Dokumente

`replaceOne(filter, replacement)`

Ersetzt ein ganzes Dokument

```
cars > db.models.updateOne(  
  { model: "Mustang Dark Horse" },  
  { $set: { price: 82000 } }  
)
```

# Update-Operatoren

| Operator        | Bedeutung           | Beispiel                                     |
|-----------------|---------------------|--|
| <b>\$set</b>    | Wert setzen         | { <b>\$set</b> : { price: 75000 } }          |
| <b>\$inc</b>    | Wert erhöhen        | { <b>\$inc</b> : { price: 1000 } }           |
| <b>\$mul</b>    | Wert multiplizieren | { <b>\$mul</b> : { price: 1.05 } }           |
| <b>\$unset</b>  | Wert entfernen      | { <b>\$unset</b> : { discount: "" } }        |
| <b>\$rename</b> | Feld umbenennen     | { <b>\$rename</b> : { oldName: "newName" } } |

# Aggregation

Komplexe Datenanalysen direkt in der Datenbank.

```
db.cars.aggregate([  
  { STUFE_1 },  
  { STUFE_2 },  
  { STUFE_3 }  
])
```

```
db.cars.aggregate([  
  { $sort: { price: -1 } },  
  { $limit: 3 }  
])
```

# Aggregationsoperatoren

| Operator               | Beschreibung                              | Beispiel   |
|------------------------|---|--|
| <code>\$match</code>   | Filtert wie <code>.find()</code>          | <code>{ \$match: { brand: „Ford“ } }</code>                                  |
| <code>\$group</code>   | Gruppiert Daten und berechnet Werte       | <code>{ \$group: { _id: "\$brand", avgPrice: { \$avg: "\$price" } } }</code> |
| <code>\$sort</code>    | Sortiert                                  | <code>{ \$sort: { price: -1 } }</code>                                       |
| <code>\$limit</code>   | Begrenzung der Ergebnisse                 | <code>{ \$limit: 5 }</code>  |
| <code>\$project</code> | Bestimmt, welche Felder ausgegeben werden | <code>{ \$project: { brand: 1, model: 1, _id: 0 } }</code>                   |

# Delete

Einzelnes Dokument löschen – `db.col.deleteOne()`

Entfernt nur das erste passende Dokument

Mehrere Dokumente löschen – `db.col.deleteMany()`

Entfernt alle passenden Dokumente

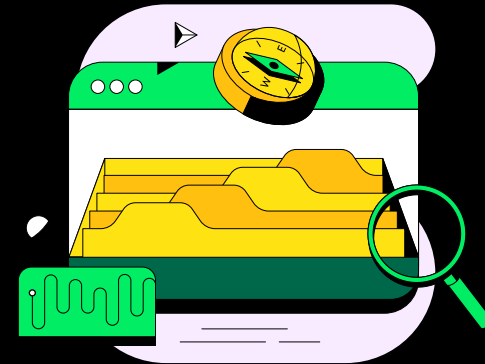
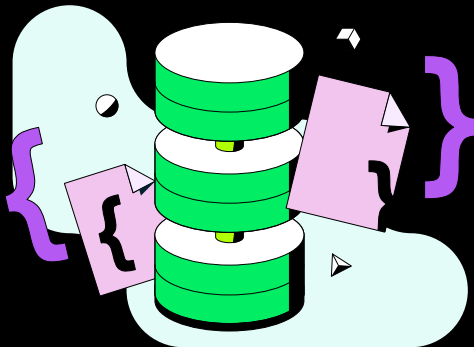
```
> db.models.deleteOne({ "brand": "tesla" })  
< {  
  acknowledged: true,  
  deletedCount: 0  
}
```



# Compass & Atlas

**MongoDB Compass** – Mächtiges grafisches Datenbankediting Werkzeug – Kostenlos

**MongoDB Atlas** – Multi Cloud basierender Datenbankservice – Zahlungspflichtig



Trompete spielend

Maus 🐭

Vielen



ören!



# Quellen

<https://de.wikipedia.org/wiki/MongoDB>

<https://theirstack.com/de/technology/mongodb>

<https://www.mongodb.com/docs/mongodb-shell/reference/data-types/>

<https://www.youtube.com/watch?v=Kk6Er0c7srU>