

# MongoDB Queries

## 1 Einfache Abfragen mit find()

### 1.1 Projektion: Welche Felder (Spalten)?

```
find({}, {_id: 0, feld1: 1, feld2: 1}) // nur feld1 und feld2 anzeigen
find({}, {_id: 0, feld1: 0, feld2: 0}) // alle Felder ausser _id, feld1 und feld2 anzeigen
```

### 1.2 Filtern: Welche Dokumente (Zeilen)?

#### 1.2.1 Vergleichsoperatoren

```
find({feld: wert}) // =
find({feld: {$eq: wert}}) // =
find({feld: {$lt: wert}}) // <
$gt // >
$gte // >=
$lte // <=
$ne // <>
```

#### 1.2.2 IN

```
find({feld: {$in: [wert1, wert2, wert3]}}) // IN
find({feld: {$nin: [wert1, wert2, wert3]}}) // NOT IN
```

```
find({feld: /A/}) // LIKE "%A%" -> Regex
```

#### 1.2.3 LIKE → Regex

```
find({"feld": /A/}) // LIKE "%A%" -> Regex
```

#### 1.2.4 Logische Operatoren

```
find({feld: {$gte: 5, $lte: 8}}) // AND
find({$and: [ {feld: {$gte: 5}}, {feld: {$lte: 8}} ]}) // AND
find({$or: [ {feld: {$lte: 5}}, {feld: {$gte: 8}} ]}) // OR
find({feld: {$not: {$gte: 5}}}) // NOT
```

### 1.3 Sortieren

```
find().sort({ feld: 1}) // ASC
find().sort({ feld: -1}) // DESC
```

### 1.4 Anzahl der Zeilen begrenzen

```
find().skip(5).limit(10) // Dokumente 6 bis 15
```

## 2 Aggregation

### 2.1 Single Purpose Aggregation

```
db.collection.find().count() // Anzahl der Dokumente
db.collection.distinct("feld") // Array mit unterschiedlichen Werten des Feldes
```

### 2.2 Aggregation Pipeline

#### 2.2.1 Grundlagen

```
db.collection.aggregate( [ { <stage1> }, { <stage2> }, ... ] )
```

#### 2.2.2 Stage: Zeilen filtern (wie bei find())

```
{ $match: { feld1: wert1 } }
```

#### 2.2.3 Stage: Spalten filtern oder hinzufügen (Projektion)

```
{
  $project: {
    _id: 0, // nicht anzeigen
    feld1: 1, // anzeigen
    "feldNeu": { $divide: ["$feld3", "$feld4"] } // berechnetes Feld
  }
}
```

#### 2.2.4 Stage: sortieren

```
{ $sort: { feld1: 1, feld2: -1 } } // 1=ASC, -1=DESC
```

#### 2.2.5 Stage: Auswertung eines Feldes über alle Dokumente

```
{
  $group: {
    _id: null,
    minimum: { $min: "$feld" },
    maximum: { $max: "$feld" },
    summe: { $sum: "$feld" },
    durchschnitt: { $avg: "$feld" }
  }
}
```

#### 2.2.6 Auswertung eines Feldes über gefilterte Dokumente ohne Gruppierung

```
db.collection.aggregate([
  { $match: { feld1: wert1 } }, // 1. Filter
  { $group: { _id: null, minimum: { $min: "$feld2" } } } // 2. Berechnung
])
```

#### 2.2.7 Gruppierung und Auswertung nach Filterung

```
db.collection.aggregate([
  { $match: { feld1: wert1 } },
  { $group: { _id: "$feld2", durchschnitt: { $avg: "$feld3" } } }
])
```

```
// 1. Filter: feld1: wert1
// 2. Gruppierung nach feld2
// 3. Auswertung von feld3
```