

MongoDB Cheat Sheet

Connection & Shell (mongosh)

MongoDB is a scalable document-oriented NoSQL database that stores JSON-like BSON records.

- **Connect:** `mongosh "mongodb://localhost:27017/mydb"`
- **List Databases:** `show dbs`
- **Select Database:** `use mydb`
- **List Collections:** `show collections`
- **Create / Drop Collection:** `db.createCollection("users") / db.users.drop()`
- **Database Stats:** `db.stats(), db.users.stats()`

Create Operations

- **Insert Single Document:**

```
db.users.insertOne({
  name: "Alice",
  email: "alice@example.com",
  age: 28,
  tags: ["dev", "db"],
  createdAt: new Date()
});
```
- **Insert Multiple Documents:**

```
db.users.insertMany([
  { name: "Bob", age: 35 },
  { name: "Charlie", age: 40 }
]);
```

Read & Query Operators

Basic Queries & Projection

- `db.users.find()` : Find all
- `db.users.findOne({ _id: ObjectId("...") })` : Find single
- **Projection:** `db.users.find({}, { name: 1, email: 1, _id: 0 })`
- **Sort,**
Skip & Limit: `db.users.find().sort({ age: -1 }).skip(10).limit(5)`

Common Query Operators

- **Comparison:** `$eq, $ne, $gt, $gte, $lt, $lte, $in: [1, 2], $nin`
- **Logical:** `$and, $or, $not, $nor`

```
db.users.find({
  $or: [{ age: { $gte: 30 } },
  { status: "VIP" } ]
});
```

- **Element & Array:**
 - Field Existence: `db.users.find({ phone: { $exists: true } })`
 - Regex Match: `db.users.find({ name: { $regex: /^Ali/ } })`
 - Array Match: `db.users.find({ tags: "dev" })`
 - Array Size: `db.users.find({ tags: { $size: 2 } })`

Update Operations & Operators

- **Update Single / Multiple:**

```
db.users.updateOne(
  { name: "Alice" },
  {
    $set: { status: "active",
    updatedAt: new Date() },
    $inc: { visits: 1 }, // Increment number
    $unset: { tempField: "" }, // Remove field
    $push: { tags: "new" }, // Append to array
    $addToSet: { tags: "unique" } // Set insertion
  },
  { upsert: true } // Create if not present
);
```
- `db.users.updateMany({ age: { $lt: 20 } }, { $set: { isMinor: true } })`

Delete Operations

- `db.users.deleteOne({ name: "Alice" })`
- `db.users.deleteMany({ status: "inactive" })`

Aggregation Pipeline

Transform and compute values across multi-stage pipelines.

```
db.orders.aggregate([
  // Stage 1: Filter
  { $match: { status: "completed" } },

  // Stage 2: Group & Aggregate
  {
    $group: {
      _id: "$customerId",
      totalSpent: { $sum: "$amount" },
      avgQuantity: { $avg: "$quantity" },
      orderCount: { $sum: 1 },
      items: { $push: "$item" }
    }
  },

  // Stage 3: Sort
  { $sort: { totalSpent: -1 } },

  // Stage 4: Limit
  { $limit: 5 }
]);
```

Key Aggregation Stages

- **\$project** : Reshape documents
- **\$unwind** : Deconstruct array field to individual documents
- **\$lookup** : Left outer join with another collection

```
{
  $lookup: {
    from: "users",
    localField: "userId",
    foreignField: "_id",
    as: "userInfo"
  }
}
```

Index Management

- **Create Index:**
`db.users.createIndex({ email: 1 }, { unique: true })`
- **Compound Index:**
`db.users.createIndex({ age: 1, name: 1 })`
- **TTL Index (Auto-Expiry):**
`db.sessions.createIndex({ createdAt: 1 }, { expireAfterSeconds: 3600 })`

- **List Indexes:** `db.users.getIndexes()`
- **Drop Index:** `db.users.dropIndex("email_1")`

Backup & Data Export Tools

- `mongodump --db=mydb --out=/backup/`
- `mongorestore --db=mydb /backup/mydb/`
- `mongoexport --db=mydb --collection=users --out=users.json`
- `mongoimport --db=mydb --collection=users --file=users.json`