

Cheat Sheet for comprehensive MongoDB

MongoDB Basics

- **Document-Oriented Storage:** MongoDB stores data in flexible, JSON-like documents.
- **Schema-less:** No fixed schema required; documents in the same collection can have different structures.
- **Scalability:** Supports horizontal scaling through sharding.
- **High Performance:** Built-in indexing, replication, and load balancing.

Installation and Setup

- **Install MongoDB:**

```
sudo apt-get install mongodb
```

- **Start MongoDB:**

```
sudo service mongodb start
```

- **Access MongoDB Shell:**

```
mongo
```

Basic Commands

- **Show Databases:**

```
show dbs
```

- **Use Database:**

```
use myDatabase
```

- **Show Collections:**

```
show collections
```

- **Create Collection:**

```
db.createCollection("myCollection")
```

- **Drop Collection:**

```
db.myCollection.drop()
```

CRUD Operations

Create

- **Insert One Document:**

```
db.myCollection.insertOne({ name: "John", age: 30 })
```

- **Insert Many Documents:**

```
db.myCollection.insertMany([
  { name: "Alice", age: 25 },
  { name: "Bob", age: 35 }
])
```

Read

- **Find All Documents:**

```
db.myCollection.find()
```

- **Find Specific Document:**

```
db.myCollection.find({ name: "John" })
```

- **Projection:**

```
db.myCollection.find({ name: "John" }, { name: 1, _id: 0 })
```

Update

- Update One Document:

```
db.myCollection.updateOne(  
  { name: "John" },  
  { $set: { age: 31 } }  
)
```

- Update Many Documents:

```
db.myCollection.updateMany(  
  { age: { $gt: 30 } },  
  { $set: { status: "Senior" } }  
)
```

Delete

- Delete One Document:

```
db.myCollection.deleteOne({ name: "John" })
```

- Delete Many Documents:

```
db.myCollection.deleteMany({ age: { $lt: 30 } })
```

Indexing

- Create Index:

```
db.myCollection.createIndex({ name: 1 })
```

- List Indexes:

```
db.myCollection.getIndexes()
```

- Drop Index:

```
db.myCollection.dropIndex({ name: 1 })
```

Aggregation Framework

- Basic Aggregation:

```
db.myCollection.aggregate([
  { $match: { age: { $gt: 30 } } },
  { $group: { _id: "$name", totalAge: { $sum: "$age" } } }
])
```

- Common Stages:

- ``$match``: Filters documents.
- ``$group``: Groups documents by a specified key.
- ``$sort``: Sorts documents.
- ``$project``: Reshapes documents.

Replication and Sharding

Replication

- Create Replica Set:

```
mongod --replSet myReplSet
```

- Initialize Replica Set:

```
rs.initiate()
```

- Add Members:

```
rs.add("host:port")
```

Sharding

- Enable Sharding:

```
sh.enableSharding("myDatabase")
```

- Shard Collection:

```
sh.shardCollection("myDatabase.myCollection", { _id: "hashed" })
```

Backup and Restore

- Backup Database:

```
mongodump --db myDatabase --out /path/to/backup
```

- Restore Database:

```
mongorestore --db myDatabase /path/to/backup/myDatabase
```

Security

- Enable Authentication:

```
mongod --auth
```

- Create User:

```
db.createUser({
  user: "myUser",
  pwd: "myPassword",
  roles: [ { role: "readWrite", db: "myDatabase" } ]
})
```

Monitoring and Performance Tuning

- Check Server Status:

```
db.serverStatus()
```

- Profile Database:

```
db.setProfilingLevel(1)
```

- Check Slow Queries:

```
db.system.profile.find().sort({ ts: -1 }).limit(10)
```

Tips and Tricks

- Use `$exists` for Field Existence:

```
db.myCollection.find({ field: { $exists: true } })
```

- **Use `\$in` for Multiple Values:**

```
db.myCollection.find({ name: { $in: ["John", "Alice"] } })
```

- **Use `\$or` for Multiple Conditions:**

```
db.myCollection.find({ $or: [{ name: "John" }, { age: 30 }] })
```

- **Use `\$lookup` for Joins:**

```
db.orders.aggregate([
  { $lookup: { from: "products", localField: "productId",
foreignField: "_id", as: "productDetails" } }
])
```

Common Errors and Solutions

- **Connection Error:**

- **Solution:** Ensure MongoDB is running and accessible.

- **Duplicate Key Error:**

- **Solution:** Ensure unique indexes are correctly set.

- **Slow Queries:**

- **Solution:** Analyze and optimize indexes and query patterns.

Advanced Features

- **Geospatial Queries:**

```
db.places.find({
  location: {
    $near: {
      $geometry: { type: "Point", coordinates: [ -73.97, 40.77 ] },
      $maxDistance: 1000
    }
  }
})
```

- **Text Search:**

```
db.articles.createIndex({ title: "text", content: "text" })  
db.articles.find({ $text: { $search: "mongodb tutorial" } })
```

Conclusion

- **Best Practices:**

- Use indexes wisely.
- Regularly backup data.
- Monitor and optimize performance.
- Secure your MongoDB instance.

This cheat sheet provides a comprehensive overview of MongoDB, covering essential commands, operations, and advanced features. Use it as a quick reference to enhance your MongoDB skills.

By Ahmed Baheeg Khorshid

ver 1.0