Important MongoDB Commands

A Quick Guide for Developers

Presented by: Adham

Email: mailinadham@gmail.com

LinkedIn: linkedin.com/in/adham-m-

Introduction

▶ What is MongoDB?

▶ - NoSQL database, document-oriented (BSON format)

▶ - Scalable, flexible, and schema-less

▶ Why Learn Commands?

▶ - Faster development, effective debugging, better performance

MongoDB Basics

▶ Start Mongo Shell: mongo

▶ Show Databases: show dbs

▶ Create/Use Database: use myDatabase

Collection Operations

- ▶ Show Collections: show collections

- ▶ Create Collection: db.createCollection("users")

- ▶ Drop Collection: db.users.drop()

CRUD - Create

- Insert One Document: db.users.insertOne({ name: "Adham", age: 25 })

- Insert Multiple Documents: db.users.insertMany([{ name: "John" }, { name: "Sara" }])

CRUD - Read

- Find One Document: db.users.findOne({ name: "Adham" })

- Find All Documents: db.users.find()

- Use Filters: db.users.find({ age: { $gt: 20 } })

CRUD - Update

- Update One Document: db.users.updateOne({ name: "Adham" }, { $set: { age: 26 } })

- Update Many Documents: db.users.updateMany({}, { $set: { active: true } })

CRUD - Delete

- Delete One Document: db.users.deleteOne({ name: "John" })

- Delete Many Documents: db.users.deleteMany({ age: { $lt: 18 } })

Indexing & Aggregation

▶ Create Index: db.users.createIndex({ name: 1 })

▶ Aggregation Example: db.users.aggregate([{ $group: { _id: "$age", count: { $sum: 1 } } }])

Useful Utilities

▶ Get Collection Stats: db.users.stats()

▶ Check Current DB: db

▶ Count Documents: db.users.countDocuments()

Conclusion

▶ MongoDB is powerful for flexible data storage.

▶ Mastering commands improves productivity.

▶ Continue exploring aggregation, performance tuning, and security.

Thanks for your attention!

▶ Presented by: Adham

▶ Email: mailinadham@gmail.com

▶ LinkedIn: linkedin.com/in/adham-m-