IMPORTANT MONGODB COMMANDS

A COMPREHENSIVE GUIDE TO MONGODB COMMANDS

Created By Selvakumar N

MONGODB

- MongoDB is another NoSQL solution
- Data is stored as BSON (Binary JSON)
- Allows storage of large amounts of data

SQL VS MONGODB

- SQL has databases, tables, rows, columns.
- Mongo has databases, collections, documents, fields.
- Both have primary keys, indexes.
- Collection structures are not enforced heavily.
- Inserts automatically create schemas.

DATABASE COMMANDS

- show dbs: Lists all available databases.
- use database_name: Creates or switches to a database.
- db.dropDatabase(): Deletes the currently selected database.

COLLECTION COMMANDS

- show collections: Lists all collections in the database.
- db.createCollection('collection_name'): Creates a new collection.
- db.collection_name.drop(): Deletes a collection.

INSERT DATA (CREATE)

- db.collection_name.insertOne({ name: 'Apple', price: 50 }): Inserts one document.
- db.collection_name.insertMany([{ name: 'Banana' }, { name: 'Mango' }]): Inserts multiple documents.

RETRIEVE DATA (READ)

- db.collection_name.find(): Retrieves all documents.
- db.collection_name.find({ name: 'Apple' }): Finds documents with a condition.
- db.collection_name.find().sort({ price: -1 }): Sorts documents by price in descending order.

MODIFY DATA (UPDATE)

- db.collection_name.updateOne({ name: 'Apple' }, { $set: { price: 60 } }): Updates one document.
- db.collection_name.updateMany({ name: 'Banana' }, { $set: { price: 25 } }): Updates multiple documents.
- db.<collection_name>.replaceOne({ <query> }, <new_document>): Replace a document

REMOVE DATA (DELETE)

- db.collection_name.deleteOne({ name: 'Apple' }): Deletes one document.
- db.collection_name.deleteMany({ name: 'Banana' }): Deletes multiple documents.

EXPORT & IMPORT DATA

- mongoexport --db=database_name --collection=collection_name --out=data.json: Exports a collection to JSON.
- mongoimport --db=database_name --collection=collection_name --file=data.json: Imports a JSON file into MongoDB.

INDEXING COMMANDS

- db.<collection_name>.createIndex({ <field>: <1 for ascending or -1 for descending> }): Create an index.
- db.<collection_name>.getIndexes(): List all indexes on a collection.
- db.<collection_name>.dropIndex(<index_name>): Drop an index

OTHER USEFUL COMMANDS

- db.<collection_name>.countDocuments({ <query> }): Count the number of documents.
- db.runCommand({ connectionStatus: 1 }): Check connection status.
- db.serverStatus(): Get server status.

CONCLUTION

- MongoDB provides powerful commands for managing databases, collections, and documents.
- The commands listed above are essential for everyday use when working with MongoDB.