mongosh MongoDB shell

CONCEPTS

mongosh is a JavaScript shell: db is the current database, db.<name> a collection, and any JS works.

Deployment server or cluster
- **Database** db / use <db>
  - **Collection** db.users
    - **Document** BSON object

CONNECT

mongosh
Connect to localhost:27017

mongosh "mongodb://<host>**:27017/**<db>**"**
Connect with a URI

mongosh --host <host> **--port** <port> **-u** <user> **-p**
Connect with options, prompt for password

--eval "<js>**"** <uri>
Run JavaScript and exit

mongosh <uri> <file.js>
Run a script file then exit

exit / quit()
Leave the shell

INSPECT

show dbs
List databases

use <db>
Switch database

db
Print the current database name

show collections
List collections

db.<coll>**.countDocuments({})**
Count matching documents

db.<coll>**.getIndexes()**
List indexes

READ

db.<coll>**.find()**
All documents (first batch)

db.<coll>**.find({** <field>**:** <value> **})**
Match on equality

db.<coll>**.findOne({** <filter> **})**
First match as one document

db.<coll>**.find({}, { **<f>**: 1 })**
Projection: pick fields

db.<coll>**.find().sort({** <f>**: -1 })**
Sort, 1 ascending, -1 descending

db.<coll>**.find().limit(**<n>**)**
Cap results; chain .skip(<n>) to page

db.<coll>**.find(**<filter>**).explain("executionStats")**
Show how the query ran

WRITE

db.<coll>**.insertOne({** <doc> **})**
Insert one document

db.<coll>**.insertMany([{ }, { }])**
Insert several documents

db.<coll>**.updateOne(**<filter>**, { $set: { **<f>**: **<v>** } })**
Change first match

db.<coll>**.deleteOne(**<filter>**)**
Delete first match

db.<coll>**.deleteMany(**<filter>**)**
Delete all matches

AGGREGATION

db.<coll>**.aggregate([**<stage>**])**
Run a pipeline of stages

{ $group: { _id: "$<f>**", n: { $sum: 1 } } }**
Group and count

INDEXES

db.<coll>**.createIndex({** <f>**: 1 })**
Create an ascending index