Basic mongo commands

use <db-name> : to use an existing DB or to create a new one if it doesn’t exist

show dbs : to view all dbs

db : for checking and accessing currently selected DB

eg. db.movie.insert({“name”:”Rocko”})

db.dropDatabase() : drop a database, delete it completely. To do so we should be in that db

db.createCollection(collection\_name, options) : normally done automatically, for example the movie collection in the above example

db.<collection\_name>.find() : to check an inserted document

eg. db.movie.find() will return

{“\_id”: ObjectId(“somenumber”), “name”: “Rocko” }

db.<collection\_name>.drop() : drop the collection, all its documents and indexes

CRUD ops

Insert

db.movie.insert([

{ name: "Avengers: Infinity War" },

{ name: "Avengers: Endgame" }

])

To insert multiple documents at a time

Embedded documents : create document containing other documents, array of documents

db.writer.insert({

writername: “Stanlee”,

comics: [

{doc1…},{doc2…},{doc3…}

]

})

db.<c\_name>.insertOne(), db.<c\_name>.insertMany() : other ways of insert op

Query ops

db.<c\_name>.find() or db.<c\_name>.pretty() : to display better

db.musicians.find({$or: [ { instrument: "Drums" }, { born: 1945 } ] }) : selectively fetch data, similar to SQL WHERE AND, WHERE OR

In the above example since multiple fields were being accessed viz. ‘instrument’ and ‘born’ hence after ‘$or:’ square brackets were used

db.musicians.find( { "instrument": { $in: [ "Keyboards", "Bass" ] } }): similar to SQL IN and NOT IN

Better way of querying with a limit

db.writers.find().pretty().limit(2) : only 2 documents displayed

Update ops

db.musicians.update(

{\_id: 4},

{

$set: { instrument: ["Vocals", "Violin", "Octapad"] }

}

)

save() : method is used to update as well as insert, a document is created if not already existing

Mongo Setup

mongod : mongo server daemon

mongo/mongosh : mongo client

Start mongod : mongod --dbpath <filepath> --bind\_ip <ip> --config <filepath>

Use mongosh : in Compass