show dbs

// Create a delete a database

use <db name>

db.dropDatabase()

// Create and delete a collection

db.createCollection(‘<collection name>’)

db.<collection name>.drop()

// Add one/many entry into a collection

db.<collection name>.insertOne({ })

db.<collection name>.insertMany({ })

// Return all documents in a collection

db.<collection name>.find()

// Filter

db.<collection name>.find({ <property name> })

ex: # this will return all the students having GPA 4.1

db.students.find({ GPA: 4.1, fullTime: false})

Compare:

db.students.find({ name: {$ne:“Bob”}}) # find all students whose names are not Bob

db.students.find({age:{$lt: 20}}) # less than

db.students.find({age:{$lte: 20}}) # less or equal than

db.students.find({age:{$gt: 20}}) # greater than

db.students.find({age:{$gte: 20}}) # greater or equal than

db.students.find({GPA: {$gte: 3.0, $lte: 4.0}) # between

in operator:

db.students.find({name: {$in: [‘Alice’, ‘Bob’]}) # return anybody whose name is Alice or Bob

db.students.find({name: {$nin: [‘Alice’, ‘Bob’]}) # return anybody whose name is not Alice or Bob

Logical operators

And:

db.students.find({$and: [{}, {}]}) # and operator, and put conditions in an array

db.students.find({$and: [{fullTime: true}, {age: {$lt: 20}}]})

or: $or

nor: $nor # both conditions need to be false

not:

db.students.find({age: {$not{$gte: 30}}}) # find students whose age is not greater than 30

Projection

If you don’t need the full object, or if you just need one field

db.students.find({<query>, <name: true>})

// Sort

db.<collection name>.find().sort({<property name>: 1 or -1})

ex: db.students.find().sort({name: 1})

// Limit the number of documents returned

db.<collection name>.find().limit(2)

// Update

db.students.updateOne({<filter>}, {<update>})

ex: db.students.updateOne({name: “Alice”}, {$set: { fullTime: true }})

db.students.updateMany({}, {}) leave the first curly braces empty, you find all documents. Then in the second document, set whatever you want to set.

Ex: db.students.updateMany({}, { $set: { fullTime: false }})

// Delete

db.students.deleteOne({ name: “Alice”})

db.students.deleteMany({ fullTime: false}) # anybody who has fullTime false is deleted

db.students.deleteMany({ registerDate: {$exists: false}}) # anybody who doesn’t have registerDate field will be deleted

Remove a field

db.students.updateOne({\_id: <id#> }, {#unset: {<field name>: “”})

// Indexing

Create index

db.students.createIndex({name: 1}) # Create index on field name with ascending order. It will return an index name

Check index

db.students.getIndexes()

Drop index

db.students.dropIndex(<index name>)