

Mongo Example

Erich Wellinger

January 16, 2017

```
1  // Start the server
2  mongod
3
4  // Start a Mongo shell
5  mongo
6
7  // Show the existing databases
8  show dbs
9
10 // Create a new database called class_db
11 use class_db
12
13 // View all the existing collections (tables)
14 db.getCollectionNames()
15
16 // Create a collection (table) and insert records into them
17 db.teachers.insert({name: 'E-Rich', age: 25, facial_hair: 'clean'})
18 db.teachers.insert({name: 'Frank', age: 21, friends: ['Adam', 'Cully']})
19 db.teachers.insert({name: 'Neil', age: 55, friends: ['Barack Obama', 'Kanye']})
20
21 // We can then view the first document in our table like so...
22 db.teachers.findOne()
23 db.teachers.findOne().pretty()
24
25 // Or we can view all our documents at once...
26 db.teachers.find().pretty()
27
28 // Or maybe we want to return the first 2 documents
29 db.teachers.find().limit(2).pretty()
30
31 // Generally speaking we run a .find({}, {}) where the first set of brackets
32 // dictates which documents get returned (like a WHERE clause) and the second
33 // set of brackets dictates which fields should be returned
34 db.teachers.find({name: 'Neil'}).pretty()
35 db.teachers.find({name: 'Neil', {friends: true}}).pretty()
36 // By default, the id is always returned but we can turn that off
37 db.teachers.find({name: 'Neil', {friends: 1, _id: false}}).pretty()
38
39 // Return only the friends entry from each document
40 db.teachers.find({}, {friends: true})
```

```

41
42 // Return all of each document (minus _id) for every document that has a friends field
43 db.teachers.find({friends: {$exists: true}}, {_id: false}).pretty()
44
45 // Return all the teachers that are under 30
46 db.teachers.find({age: {$lte: 30}}).pretty()
47
48 // Let's add up the ages of Neil and Frank
49 db.teachers.aggregate([
50   { $match: { name: {$in : ['Frank', 'Neil']} } },
51   { $group: { _id: "$name", something: { $sum: "$age" } } },
52   { $sort: { something: 1 } }
53 ])
54
55 // Let's update Neil's facial hair status
56 db.teachers.update({name: "Neil"}, {$set : {facial_hair: 'Jesus Beard'}},
57   {multi: true, upsert: true})
58
59 // Let's kick it Old-School and update Frank's name to reflect that
60 db.teachers.find({name: "Frank"}).
61   forEach(function(doc) {
62     {
63       id = doc._id;
64       new_name = doc.name.replace("Frank", "Frank-The-Tank");
65       db.teachers.update({ _id: id },
66         {$set: {name: new_name}})
67     };
68   });
69
70 // Drop a particular collection
71 db.teachers.drop()

```