



Introduction to Database Systems: CS312

Editing Documents

Oliver Bonham-Carter

3 April 2019

Getting started with Mongo

Start
MongoDB

Modifying
Documents

Setup a data directory (if you have not already done so)

```
mkdir ~/mongodbData
```

Start the Mongo server with data directory as a parameter

Note: Control-C to exit.

```
mongod --dbpath ~/mongodbData/
```

With new another terminal, start the Mongo client

```
mongo
```

Find databases or collections, from Mongo's client

```
show dbs
```

```
show collections
```

Begin a new database, from Mongo's client

```
use myDB
```

Adding to a Document in Mongo With a Date

Start
MongoDB

Modifying
Documents

Insertion of several documents using insertMany()

```
db.restaurants.insertMany( [  
    { "_id" : 1, "name": "Central Perk Cafe", "Borough": "Manhattan" },  
    { "_id" : 2, "name": "Rock A Feller Bar and Grill", "Borough": "Queens", "violations": 2 },  
    { "_id" : 3, "name": "Empire State Pub", "Borough": "Brooklyn", "violations": 0 },  
    { "_id" : 4, "name": "The Captain's Cafe", "Borough": "London" },  
    { "_id" : 5, "name": "The Resto in a Cave", "Borough": "Paris", "violations": 0 }  
    { "_id" : 6, "name": "The Crow Bar", "Borough": "Paris", "violations": 98, "note": "gross"}  
]);
```

- Note: we are able to override the `_id` settings from Mongo and implement our own values.

Start
MongoDB

Modifying
Documents

Add to document

```
db.restaurants.updateOne(  
  {"name": "Empire State Pub"},  
  { $set: {  
    rating: "Thumbs-Up",  
    status: "Loved it!" },  
    $currentDate: { lastModified: true }  
});
```

```
{  
  "_id" : 3,  
  "name" : "Empire State Pub",  
  "Borough" : "Brooklyn",  
  "violations" : 0,  
  "lastModified" : ISODate("2019-04-03T01:06:48.584Z"),  
  "rating" : "Thumbs-Up",  
  "status" : "Love it!"  
}
```

Inserting New Data to Existing Document

Start
MongoDB

Modifying
Documents

```
db.restaurants.updateOne(  
  {"name": "Central Perk Cafe"},  
  { $set: {  
    rating: "Good",  
    status: "Loved it!",  
    kitchenQuality: "ok" },  
    $currentDate: { lastModified: true }  
});
```

```
> db.restaurants.find({name:"Central Perk Cafe"},{}).pretty()  
{  
  "_id" : 1,  
  "name" : "Central Perk Cafe",  
  "Borough" : "Manhattan",  
  "kitchenQuality" : "ok",  
  "lastModified" : ISODate("2019-04-03T01:51:11.879Z"),  
  "rating" : "Good",  
  "status" : "Loved it!"  
}
```

Modifying

Start
MongoDB

Modifying
Documents

Add, Edit document

```
db.restaurants.updateOne(  
  {"name": "The Resto in a Cave"},  
  {  
    $set: {  
      rating: "Musty",  
      status: "Dirty and full of rocks!!!",  
      kitchenQuality: "A fire in the floor",  
      Note: "The waiter was a bear who chased me." },  
    $currentDate: { lastModified: true }  
  });
```

```
> db.restaurants.find({name: "The Resto in a Cave"}, {}).pretty()  
{  
  "_id" : 5,  
  "name" : "The Resto in a Cave",  
  "Borough" : "Paris",  
  "violations" : 0,  
  "Note" : "The waiter was a bear who chased me.",  
  "kitchenQuality" : "A fire in the floor",  
  "lastModified" : ISODate("2019-04-03T02:03:59.055Z"),  
  "rating" : "Musty",  
  "status" : "Dirty and full of rocks!!!"  
}
```

Cool! Let's See That in Action!

Start
MongoDB

Modifying
Documents



BTW: This graphic was *also* found as a result of image searching,
“The Coolest Thing.” Apologies (again!)

Consider this...

Start
MongoDB

Modifying
Documents

THINK

- Take about five minutes to play with code!
 - Try adding new attributes to a document
 - Try modifying existing attributes in a document
 - Add some numerical data to try the encoded operators;
 $\{<, >, \leq, \geq, ==\}$ (see next slide for code)



How Do I Compare Values?

From last time ...

Start
MongoDB

Modifying
Documents

Operation	MongoDB	RDBMS
Equality	db.employees.find({"salary": "5000"})	where 'salary' = '5000'
Less Than	db.employees.find({"age": {\$lt: 30}})	where age < 30
Less Than Equals	db.employees.find({"age": {\$lte: 30}})	where age <= 30
Greater Than	db.employees.find({"age": {\$gt: 30}})	where age > 30
Greater Than Equals	db.employees.find({"age": {\$gte: 30}})	where age >= 30
Not Equals	db.employees.find({"age": {\$ne: 30}})	where age != 30

Details where age less than 30

```
db.employee.find({"age": {$lt: 30}}, {"email": 1}).pretty()
```

```
> db.employee.find({"age": {$lt: 30}}, {"email": 1}).pretty()
{
    "_id" : "5c89c73ec4863a49668b3784",
    "email" : "belinda.guzman@beadzza.org"
}
{ "_id" : "5c89c73e5e7b4f23f0a32660", "email" : "cora.keith@combogen.io" }
```

Consider this...

Start
MongoDB

Modifying
Documents

THINK

- Can you create and populate a completely new MongoDB database?
- Can you write sophisticated queries in your database to isolate meaningful information from the data?