Document Databases: MongoDB

# Install MongoDB

Create a MongoDB repository config file.

$ sudo vim /etc/yum.repos.d/mongodb-org-3.4.repo

Switch vim into insert mode with pressing “i” and copy following into it.

[mongodb-org-3.4]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.4/x86\_64/

gpgcheck=1

enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc

Exit insert mode with **Esc** and save the file with **:wq** and pressing **Enter**.  
Check whether repo is added correctly.

$ yum repolist

…

mongodb-org-3.4 MongoDB Repository

…

Install main MongoDB package.

$ sudo yum install -y mongodb-org

Run the server.

$ sudo service mongod start

Perform a sanity check.

$ mongo

> db.version()

> db.help()

> exit

# Add the data

We’ll be using a sample dataset describing restaurants. Download it:

$ wget https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json

If you wish, check its contents. Every line contains a JSON object describing a restaurant, including scores it was given by visitors.

Import the data into database “lab”, collection “restaurants” which will be created implicitly.

$ mongoimport --db lab --collection restaurants --drop --file primer-dataset.json

Switch to lab db in mongo shell.

$ mongo

> use lab

Output one of the records.

> db.restaurants.findOne()

Output 20 of the records with less formatting.

> db.restaurants.find()

Copy and paste a screenshot or text of the output for the above commands here:



# Try queries

Let’s write some queries that refine the results. Find a restaurant in Manhattan borough.

> db.restaurants.findOne({"borough": "Manhattan"})

Find more restaurants in Manhattan, but output only name of the restaurant, its street and building.

> db.restaurants.find({"borough": "Manhattan" }, {"\_id": 0, "name": 1, "address.street": 1, "address.building": 1 })

Display name and type of cuisine for 10 restaurants in Manhattan that are not Italian or Irish and have not been graded.

> db.restaurants.find({"borough": "Manhattan", $nor: [{"cuisine": "Italian"}, {"cuisine": "Irish"}], "grades.score": {$exists: false}}, {"cuisine": 1, "name": 1, "\_id": 0}).limit(10)

You might notice that one of the records returned by previous query is following.

{ "cuisine" : "Other", "name" : "Test" }

You can change the cuisine type of that restaurant like this.

> db.restaurants.update({name: "Test"}, {$set: {cuisine: "Italian"}})

Similarly, you can remove a record.

> db.restaurants.remove({"name" : "Biters Llc"})

Pick one restaurant of your own choosing. Look up 5 restaurants on the same street but with different cuisine, output only their names and scores. If you can output an average score, you are a hero, but don’t spend too much time, separate grades are okay.

Paste your query and results here:

