NoSQL

Objectives

- Introduction to NoSQL and MongoDB
- Web scraping with Cheerio
- MongoJS

MongoDB

What’s a SQL database?

What’s the structure of a SQL database?

What’s a NoSQL database?

A mechanism for storage and retrieval of data that is modeled in means other than tabular relations

RTFM https://en.wikipedia.org/wiki/NoSQL

MongoDB

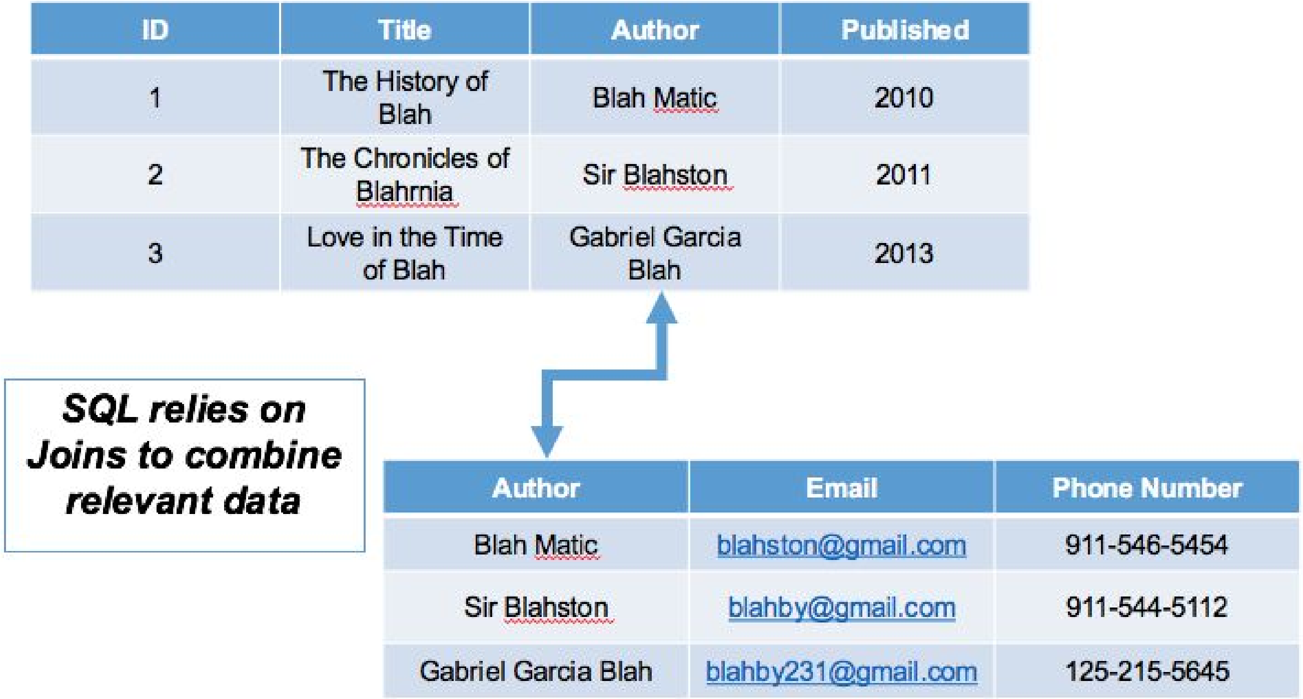
MongoDB is a very popular noSQL Database

It uses a document-oriented model as opposed to a table-based relational model (SQL)

MongoDB stores data in BSON Format (effectively compressed JSONs)

MongoDB has tons of drivers and packages for connecting to Node, C++, Java, etc.

Relational Database

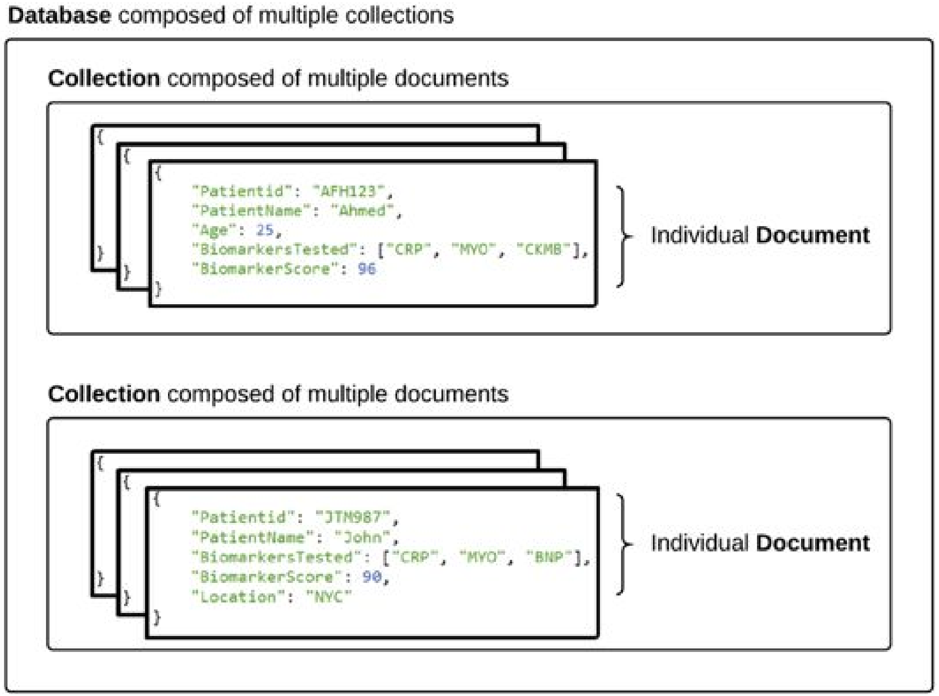


ID	Title	Author	Published
1	The History of Blah	Blah Matic	2010
2	The Chronicles of Blahrnia	Sir Blahston	2011
3	Love in the Time of Blah	Gabriel Garcia Blah	2013

Author	Email	Phone Number
Blah Matic	blahston@gmail.com	911-546-5454
Sir Blahston	blahby@gmail.com	911-544-5112
Gabriel Garcia Blah	blahby231@gmail.com	125-215-5645

SQL relies on Joins to combine relevant data

Document Database


Database composed of multiple collections
Collection composed of multiple documents
Individual **Document**
Collection composed of multiple documents
Individual **Document**

```
{
    "Patientid": "AFH123",
    "PatientName": "Ahmed",
    "Age": 25,
    "BiomarkersTested": ["CRP", "MYO", "CKMB"],
    "BiomarkerScore": 96
}
```

```
{
    "Patientid": "JTN987",
    "PatientName": "John",
    "BiomarkersTested": ["CRP", "MYO", "BNP"],
    "BiomarkerScore": 90,
    "Location": "NYC"
}
```

RTFM https://www.mongodb.com/document-databases

SQL	NoSQL
Database	Database
Table	Collection
Row	Document
Column	Field

Activity: Research NoSQL && MongoDB

Work with your neighbors to research the following:

- What are the advantages of using a noSQL database like MongoDB according to the MongoDB Website?
- What are the advantages of using a noSQL database like MongoDB according to the web (places like Quora)?
- What are the disadvantages of using a noSQL database like MongoDB according to the web (places like Quora)?

What are the advantages (according to MongoDB)?

- Relational databases require that schemas be defined before you can add data. For example, you might want to store data about your customers such as phone numbers, first and last name, address, city and state – a SQL database needs to know what you are storing in advance.
- Object-oriented programming that is easy to use and flexible.

OOP?

RTFM https://en.wikipedia.org/wiki/Object-oriented_programming

&& https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_JS

What are the advantages (according to the “experts”)?

What are the disadvantages (according to the “experts”)?

Activity: Install MongoDB (20 min)

See Installing-MongoDB.md

mongod

Demo: MongoDB Basics

See 01-Create-Insert-and-Find

- $ mongod
- Leave it running; open a new Terminal
- $ mongo

```
use <database name>
```

We use use to not only switch between databases, but to create a new database

But. Our new database doesn’t exist until we add a collection db

In the mongo shell, db is the variable that references the current database.

show dbs

admin && local

RTFM https://docs.mongodb.com/manual/reference/local-database/
&& https://docs.mongodb.com/manual/reference/system-collections/

```
db.<collection-name>.insert()
```

```
{
"continent": "Africa",
    "country":"Morocco",
    "majorcities": [
        "Casablanca",
        "Fez",
        "Marrakech"]
}
```

```
db.<collection-name>.find()
db.<collection-name>.pretty()
```

show collections

Activity: MongoDB Basics (15 min)

See 02-Starting-With-Mongo

Review: MongoDB Basics

Somebody Slack their solution to #class-instruction

$in

The $in operator selects the documents where the value of a field equals any value in the specified array.

```
db.classroom.find({"hobbies": {$in: ["Coding"]}})
```

RTFM https://docs.mongodb.com/manual/reference/operator/query/in/

Demo: Update, Remove and Drop

```
use lessondb
```

```
db.iWantToGoToThere.update({"country": "Morocco"}, {$set: {"continent":"Antarctica"}})
db.<collection-name>.update({“field”: “value”}, {$set: {“field”: “value”}})
```

$set

The $set operator replaces the value of a field with the specified value.

RTFM https://docs.mongodb.com/manual/reference/operator/update/set/

What about updating multiple entries?

{multi:true}

```
db.iWantToGoToThere.update({"country": "Morocco"}, {$set: {"continent":"Antarctica"}},
{multi:true})
```

What will happen when we run the following command?

```
db.places.update({"country": "Morocco"}, {$set: {"capital":"Rabat"}})
```

The field is created for us.

$push

The $push operator appends a specified value to an array.

```
db.places.update({"country": "Morocco"}, {$push: {"majorcities":"Agadir"}})
db.places.remove({"country":"Morocco"})
db.<collection-name>.remove()
```

Removes documents from a collection

```
db.<collection-name>.drop()
```

Removes a collection from a database

```
db.dropDatabase()
```

Activity: Taking Candy from Basket Weavers(20 min)

See 04-Update-Delete-and-Drop

Review: Taking Candy from Basket Weavers

Somebody Slack their solution to #class-instruction

Activity: Robomongo aka Robo 3T (5 min)

https://robomongo.org/download

If you don't have one, create a new connection for localhost

Demo: Robo 3T

Activity: Robo 3T (and only Robo 3T!) (10 min)

- Drop the classroom collection and create a new one
- In a new classroom collection, reenter your name, os, and hobby info array. This should be entered using the right-click -> Insert Object method.
- Next, Slack out your name, OS and hobbies into the classroom chat.
- As student info comes in, add it into the database.
- As students enter their BSON info into slack, insert it into your database.
- By the end of the exercise, you should have every student's information in your classroom collection.

How do you learn more about the Mongo shell?

help

How do you exit the Mongo shell?

exit || quit()

Lunch?

What is web scraping?

A programmatic means of taking data from any web site.

What are some reasons we might want to scrape data?

Do you think there might be legal ramifications for scraping certain websites?

Cheerio

https://github.com/cheeriojs/cheerio

Demo: Web Scraping

See 05-Scraping

- $ npm install
- https://www.reddit.com/r/webdev/
- $ node server.js
- Watch the magic!

Review: server.js

- Note:
  - the use of the "request" package to take in the HTML body
  - how Cheerio loads the HTML data and uses the $ as a var.
  - Why do you think we would use that symbol?
    - tto give it the same syntax as jQuery for selecting HTML elements
  - how we push the information into an empty array
  - how Cheerio's .children method can be used to select child elements within a selected element

Activity: Scrape

Run server2.js and server3.js

Discuss with your neighbor

Be prepared to report out on what you discover

P.S.: The awwwards.com site has since changed.

Can you debug server3.js to scrape image URLs?

Activity: My First Web Scraper (15 min)

See 06-Scrape-Starter

Students: Using this template, the cheerio documentation, and what you've learned in class so far, scrape a website of your choice, save information from the page in a result array, and log it to the console.

Review

Slack your scrapers to #class-instruction

Issues

- Not all data is available just from requesting the HTML. Some HTML is generated client-side after the initial page load.
    - With scraping this way, you can only get the initial HTML sent from the server. Any HTML added with JavaScript won't be accessible via scraping.
    - You can check to see if the data will be available by disabling JavaScript in the browser and reloading the page (be sure to re-enable afterwards!).

Activity: Meanwhile, Back at MongoDB… (5 min)

See Slack for instructions

Demo: Animals

07-Insert Animals

Demo: Sorting

use zoo

Sort by id:

- db.animals.find().sort({_id:1})
- db.animals.find().sort({_id:-1})

Sort by integer

- db.animals.find().sort({numlegs:1})
- db.animals.find().sort({numlegs:-1})

Sort by string:

- db.animals.find().sort({class:1})
- db.animals.find().sort({class:-1})

Ascending: .sort({[field]:1})

Descending: .sort({[field]:-1})

MongoJS

https://www.npmjs.com/package/mongojs

Activity: RTFM

How would you use the following Mongo methods with MongoJS?

- find
- insert
- remove
- sort

Demo: MongoDB and Sorting

```
$ node 09-MongoDB-and-Sorting/Solved/server.js
```

http://localhost:3000/

http://localhost:3000/all

http://localhost:3000/name

http://localhost:3000/weight

Activity: MongoDB & Sorting (20 min)

See 09-MongoDB-and-Sorting

TODO: Make four routes that display results from your zoo collection

0: Root: Displays a simple "Hello World" message (no mongo required).
1: All: Send JSON response with all animals
2: Name: Send JSON response sorted by name in ascending order
3: Weight: Send JSON response sorted by weight in descending order

Review: MongoDB & Sorting

What’s missing from the full-stack?

Demo: MongoJS + The Front-End

```
$ node 10-MongoJS-and-the-Front-End/server.js
```

http://localhost:3000/

Note how it pulls in the information we typed in earlier for our zoo collection, and places it in an HTML table.

Activity: MongoJS + The Front-End (25 min)

See 10-MongoDB-and-the-Front-End

1. Make an AJAX function for loading the table in index.html with the data from your animals collection in MongoDB. Each row should have info for one animal.
2. Make two AJAX functions that fire when users click the two buttons on index.html.
   a. When the user clicks the Weight button, the table should display the animal data sorted by weight.
   b. When the user clicks the Name button, the table should display the animal data sorted by name.

Activity: Scrape to a Database (40 min)

See 11-Scraping-into-a-db

Using the tools and techniques you learned so far, you will scrape a website of your choice, then place the data in a MongoDB database. Be sure to make the database and collection before running this exercise.

Consult the assignment files from earlier in the class if you need a refresher on Cheerio.

Review: Scrape to a Database

- Each route starts with the usual Express method (app.get, app.post, etc.).
- The route that retrieves the data is simply a MongoJS call wrapped in a get.
- The scrape route consists of a cheerio call to scrape the site enclosing a MongoJS method that saves the site data to the server.

Further Reading

- https://en.wikipedia.org/wiki/Robots_exclusion_standard