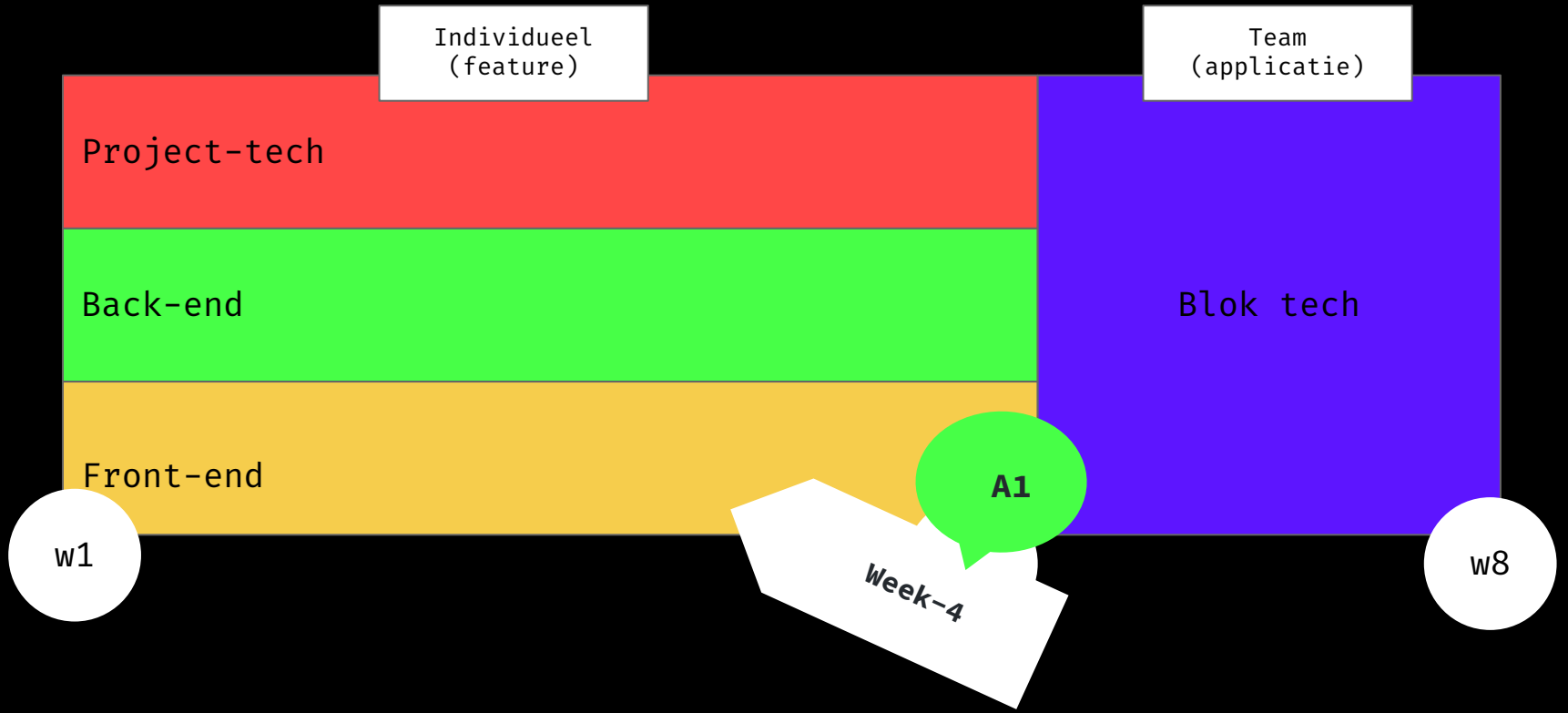
back-end

Database

lab 4/8

Stand-up!

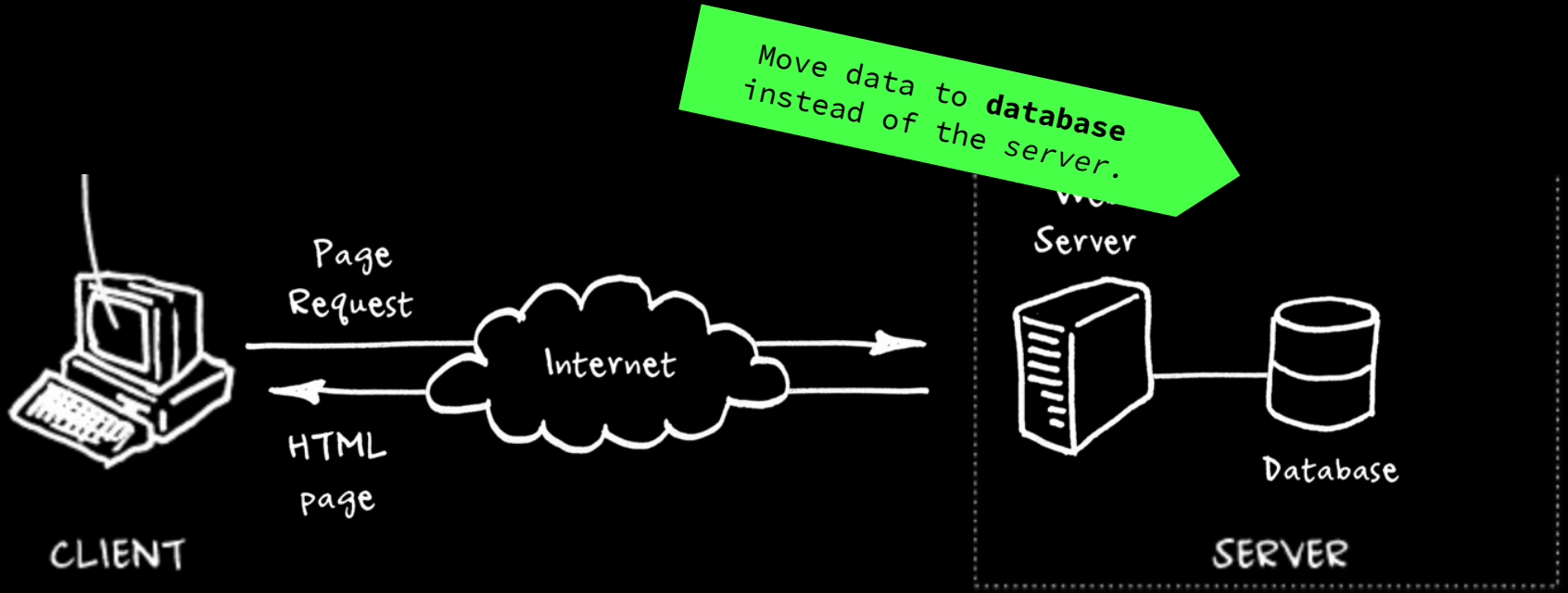
Show what you did


Individueel
(feature)
Team
(applicatie)
Project-tech
Back-end
Front-end
Blok tech
A1
Week-4
w1
w8

today

I. ~~Stand-up~~
II. Connect (recap)
III. SQL vs NoSQL
IV. Crud

Connect


Move data to **database** instead of the server.
Page Request
HTML page
Internet
Server
Database
CLIENT
SERVER

connect

mongodb

MongoDB (from hu**mong**ous) is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. MongoDB is developed by MongoDB Inc. […]

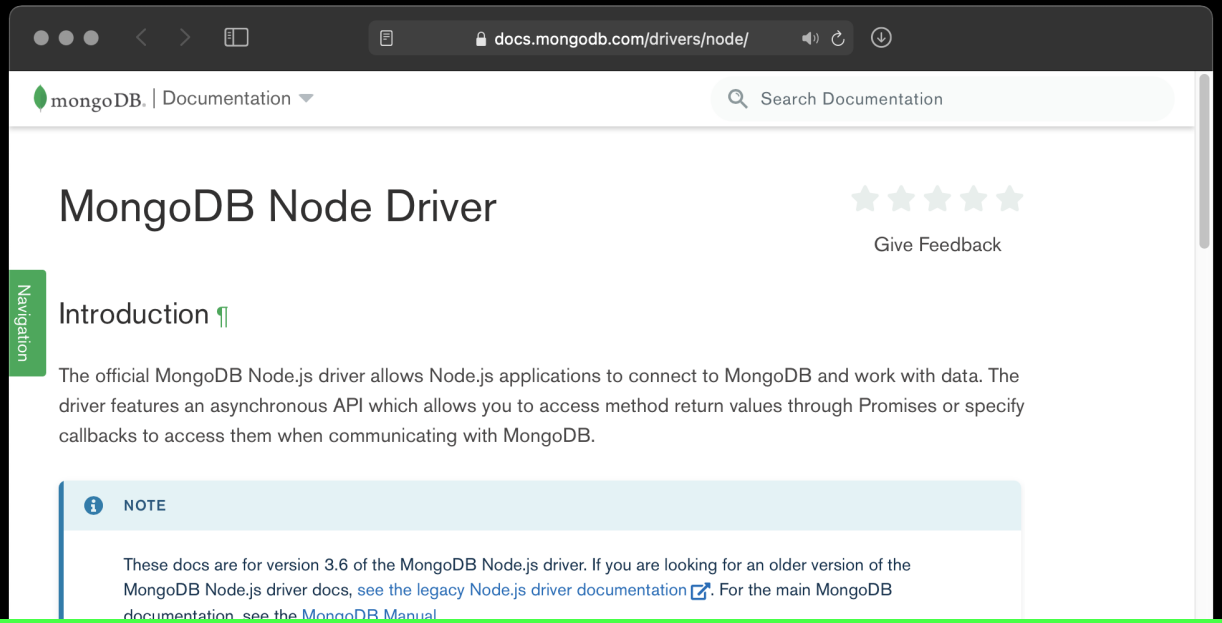
wikipedia.org

connect

mongodb

- JavaScript can be used in queries, aggregation function
- Map-reduce can be used for batch processing of data and aggregation operations
- Manage massive increases in new, rapidly changing data types

https://www.mongodb.com/compare/mongodb-mysql

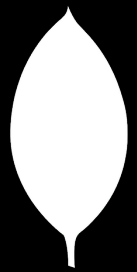


Note: there are a lot of small steps involved. Read the Mongo guides very carefully. If you miss a step everything will be broken.

bash

```
$ npm install mongodb d

+ mongodb@2.2.33
+ dotenv@4.0.0
added 11 packages in 4.022s

$
```

mongodb wraps **MongoDB** for **Node**



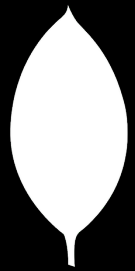
connect

mongodb

```
// Files
mongodb-server/
├─ node_modules/
├─ static/
│  ├─ index.css
│  ├─ index.js
│  └─ upload/
├─ view/
│  ├─ add.ejs
│  ├─ detail.ejs
│  ├─ head.ejs
│  ├─ list.ejs
│  ├─ not-found.ejs
│  └─ tail.ejs
├─ .env
├─ index.js
└─ package.json
```

.env

```
DB_HOST=localhost
DB_PORT=27017
DB_NAME=mymoviewebsite
DB_USERNAME=dandevri
```



connect

mongodb

```
// Files
mongodb-server/
├─ node_modules/
├─ static/
│  ├─ index.css
│  ├─ index.js
│  └─ upload/
├─ view/
│  ├─ add.ejs
│  ├─ detail.ejs
│  ├─ head.ejs
│  ├─ list.ejs
│  ├─ not-found.ejs
│  └─ tail.ejs
```

.env

```
DB_HOST=localhost
DB_PORT=27017
DB_NAME=mymoviewebsite
```

Note: Never ever put your **host and password in code or on GitHub**! People will be able to access your database!

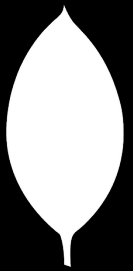
connect

mongodb

```
// Files
mongodb-server/
├─ node_modules/
├─ static/
│  ├─ index.css
│  ├─ index.js
│  └─ upload/
├─ view/
│  ├─ add.ejs
│  ├─ detail.ejs
│  ├─ head.ejs
│  ├─ list.ejs
│  ├─ not-found.ejs
│  └─ tail.ejs
├─ .env
├─ .gitignore
├─ index.js
└─ package.json
```

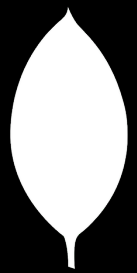
.gitignore

```
node_modules/
.DS_Store
.env
```



index.js

```
…
var multer = require('multer')
var mongo = require('mongodb')

require('dotenv').config()

var db = null
var url = 'mongodb://' + process.env.DB_HOST + ':' +
process.env.DB_PORT

mongo.MongoClient.connect(url, function (err, client) {
  if (err) throw err
  db = client.db(process.env.DB_NAME)
})

…
```



```
function add(req, res, next) {
  db.collection('movie').updateOne({
      _id: ObjectID(req.body._id),
      {$set: {textProfile: req.body.description}}
  }, done)

  function done(err, data) {
    if (err) {
      next(err)
    } else {
      res.redirect('/' + data.insertedId)
    }
  }
```

live demo .env en connection

SQL & NoSQL

sql

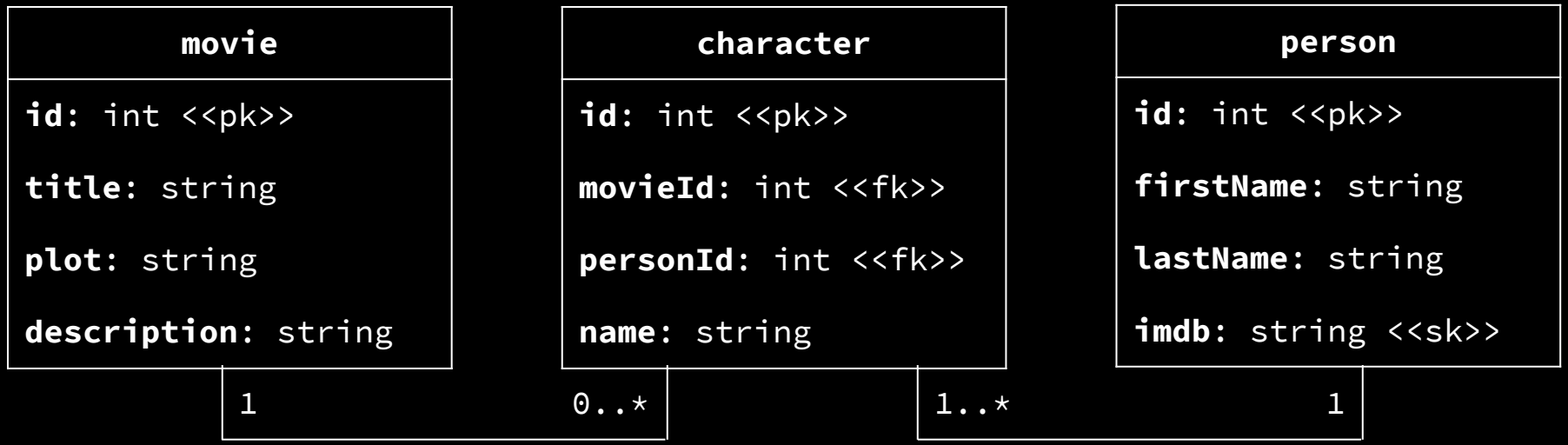
?

SQL (Structured Query Language) is a […] language used in programming and designed for managing data held in a relational database […].

wikipedia.org

sql

structure



nosql ?

A NoSQL (originally referring to “non SQL” or “non relational”) database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases. […]

wikipedia.org

nosql

?

[…] Such databases have existed since the late 1960s, but did not obtain the “NoSQL” moniker until a surge of popularity in the early twenty-first century, triggered by the needs of Web 2.0. […]

wikipedia.org

mongodb

?

MongoDB (from hu**mong**ous) is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. MongoDB is developed by MongoDB Inc. […]

wikipedia.org

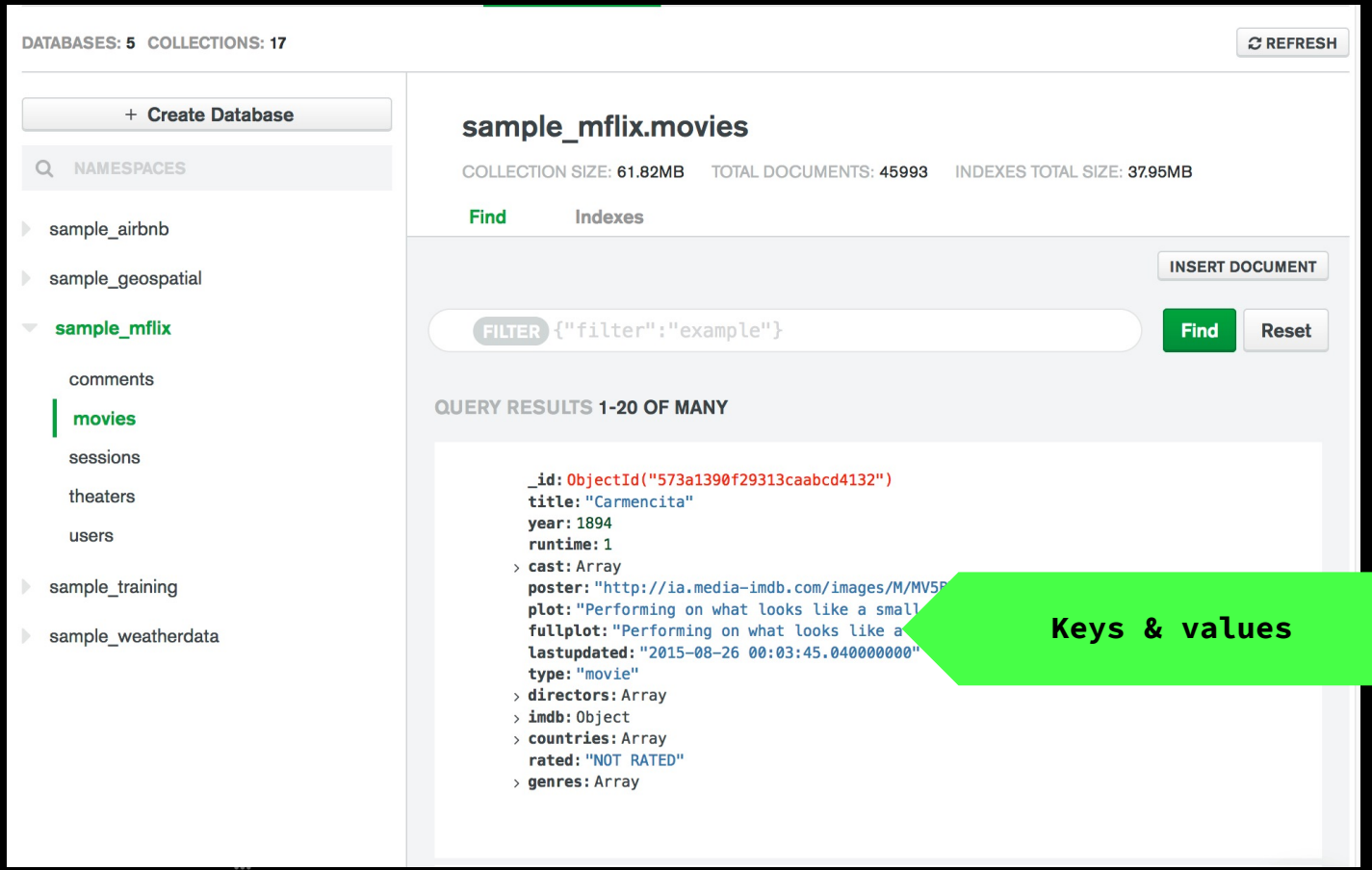
mongodb

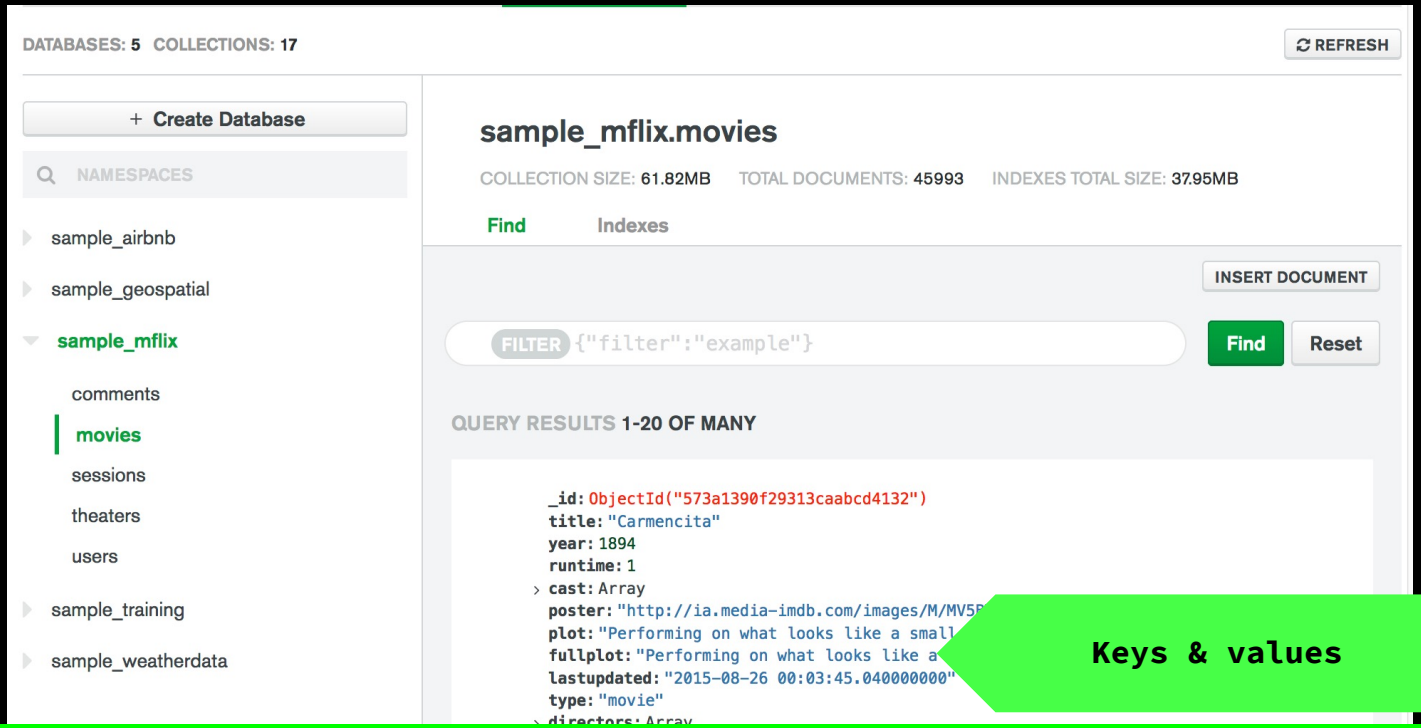
?

MongoDB (from hu**mong**ous) is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. Mo[illegible] is developed by MongoDB Inc. […]

JSON like **documents** (no relations)

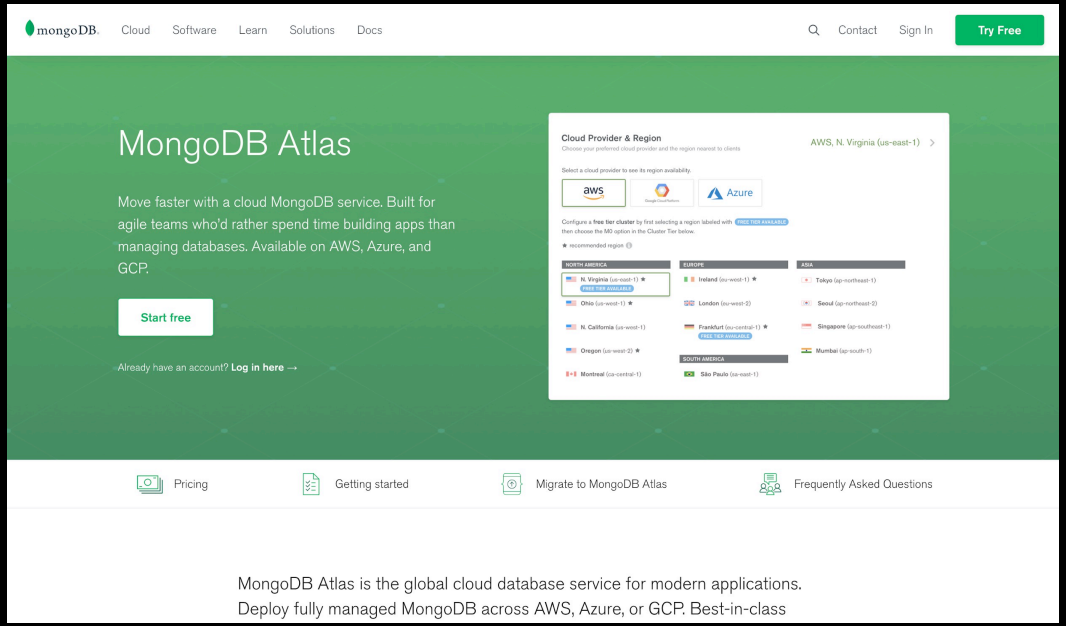
wikipedia.org


DATABASES: 5 COLLECTIONS: 17
REFRESH
+ Create Database
NAMESPACES
sample_airbnb
sample_geospatial
sample_mflix
comments
movies
sessions
theaters
users
sample_training
sample_weatherdata
sample_mflix.movies
COLLECTION SIZE: 61.82MB
TOTAL DOCUMENTS: 45993
INDEXES TOTAL SIZE: 37.95MB
Find
Indexes
INSERT DOCUMENT
FILTER {"filter":"example"}
Find
Reset
QUERY RESULTS 1-20 OF MANY
_id: ObjectId("573a1390f29313caabcd4132")
title: "Carmencita"
year: 1894
runtime: 1
cast: Array
poster: "http://ia.media-imdb.com/images/M/MV5
plot: "Performing on what looks like a small
fullplot: "Performing on what looks like a
lastupdated: "2015-08-26 00:03:45.040000000"
type: "movie"
directors: Array
imdb: Object
countries: Array
rated: "NOT RATED"
genres: Array
Keys & values



Think about how you want to structure you data.
This is called data modelling.

mongodb atlas



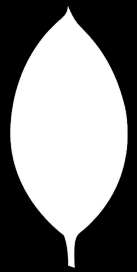
https://www.mongodb.com/cloud/atlas

Crud

bash

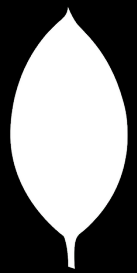
```
$ npm install mongodb d

+ mongodb@2.2.33
+ dotenv@4.0.0
added 11 packages in 4.022s

$
```

mongodb wraps **MongoDB** for **Node**



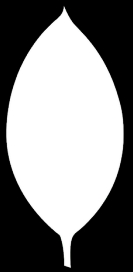
index.js

```
…
var multer = require('multer')
var mongo = require('mongodb')

require('dotenv').config()

var db = null
var url = 'mongodb://' + process.env.DB_HOST + ':' +
process.env.DB_PORT

mongo.MongoClient.connect(url, function (err, client) {
  if (err) throw err
  db = client.db(process.env.DB_NAME)
})

…
```



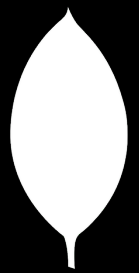
index.js

```
…

function movies(req, res, next) {
  db.collection('movie').find().toArray(done)

  function done(err, data) {
    if (err) {
      next(err)
    } else {
      res.render('list.ejs', {data: data})
    }
  }
}

…
```



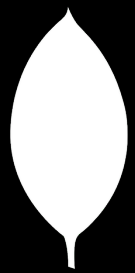
index.js

```
…
function movie(req, res, next) {
  var id = req.params.id
  db.collection('movie').findOne({
    _id: mongo.ObjectID(id)
  }, done)

  function done(err, data) {
    if (err) {
      next(err)
    } else {
      res.render('detail.ejs', {data: data})
    }
  }
}

…
```



index.js

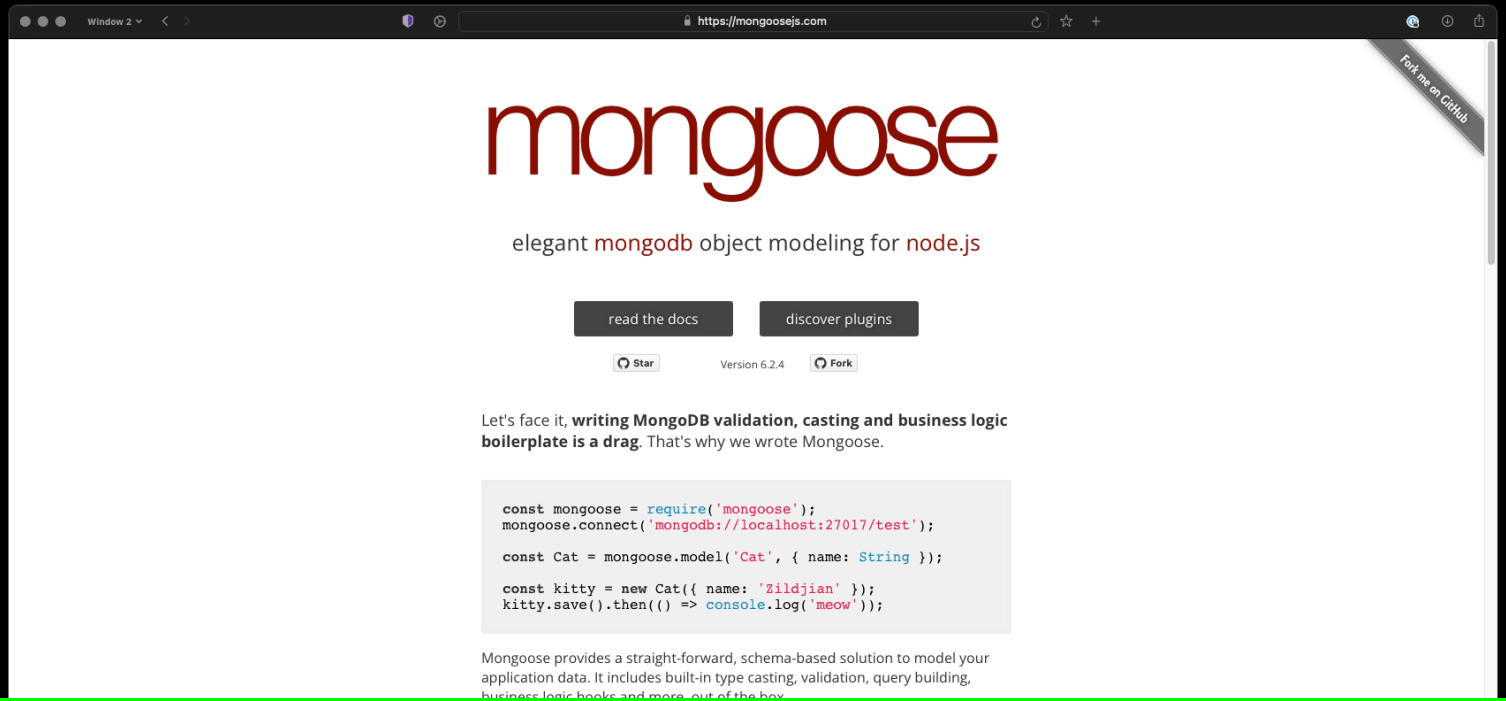
```
…

function add(req, res, next) {
  db.collection('movie').updateOne({
          _id: ObjectID(req.body._id),
          {$set: {textProfile: req.body.description}}
  }, done)

  function done(err, data) {
    if (err) {
      next(err)
    } else {
      res.redirect('/' + data.insertedId)
    }
  }
}
```



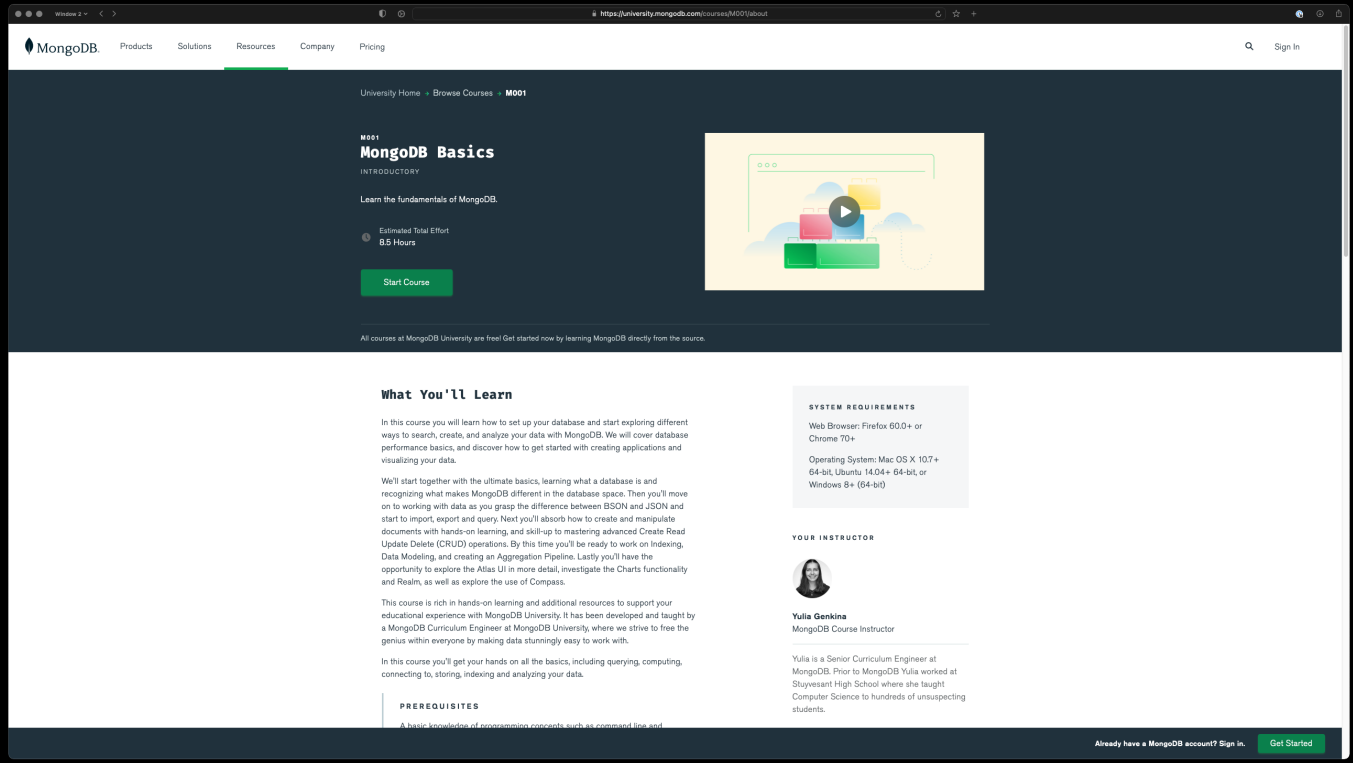
index.js

```
…
function remove(req, res, next) {
  var id = req.params.id
  db.collection('movie').deleteOne({
    _id: mongo.ObjectID(id)
  }, done)

  function done(err) {
    if (err) {
      next(err)
    } else {
      res.json({status: 'ok'})
    }
  }
}

…
```

Note: for A1 to pass you'll need to find and update



Pick mongoDB (default driver) over Mongoose.



MongoDB University

```
function add(req, res, next) {
  db.collection('movie').updateOne({
      _id: ObjectID(req.body._id),
      {$set: {textProfile: req.body.description}}
  }, done)

  function done(err, data) {
    if (err) {
      next(err)
    } else {
      res.redirect('/' + data.insertedId)
    }
  }
```

live demo crud operations

exit;

see you in lab-5!