

# back-end

## Database

lab-4a

**Stand-up!**

*Show what  
you did*

# today

~~I. Stand up~~

II. SQL vs NoSQL

III. Crud

IV. Sessions

# SQL & NoSQL

# sql

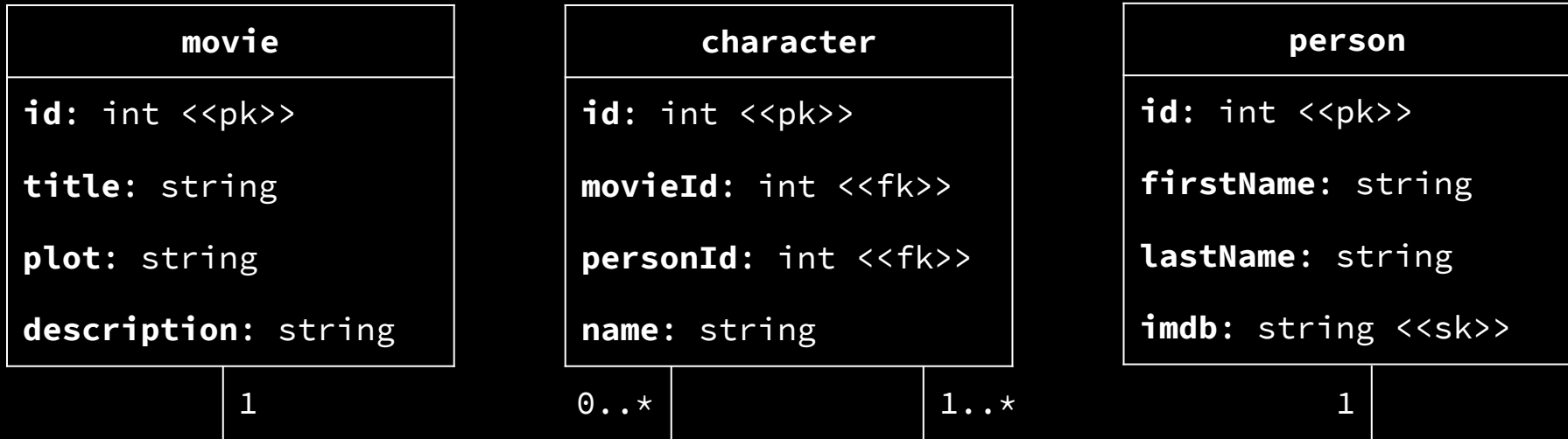
?

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

[wikipedia.org](https://en.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](https://en.wikipedia.org/wiki/NoSQL)

# 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](https://en.wikipedia.org)



# mongodb

?

MongoDB (from hum**ong**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](https://www.wikipedia.org)

# mongodb

?

MongoDB (from humongous) 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. [...]

JSON like documents (no relations)

[wikipedia.org](https://www.wikipedia.org)

DATABASES: 5 COLLECTIONS: 17

REFRESH

+ Create Database

Q 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/MV5B
  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



# mongodb

# atlas

The screenshot shows the MongoDB Atlas website. The header includes the MongoDB logo, navigation links (Cloud, Software, Learn, Solutions, Docs), a search icon, and links for Contact, Sign In, and a green 'Try Free' button. The main content area has a green background with the text 'MongoDB Atlas' and a description: 'Move faster with a cloud MongoDB service. Built for agile teams who'd rather spend time building apps than managing databases. Available on AWS, Azure, and GCP.' Below this is a 'Start free' button and a link for existing users. A white modal window titled 'Cloud Provider & Region' is open, showing options for AWS, Google Cloud, and Azure. It lists recommended regions for each provider, such as N. Virginia for AWS and Ireland for Google Cloud. The footer contains links for Pricing, Getting started, Migrate to MongoDB Atlas, and Frequently Asked Questions. At the bottom, a text block states: 'MongoDB Atlas is the global cloud database service for modern applications. Deploy fully managed MongoDB across AWS, Azure, or GCP. Best-in-class'.

mongodb Cloud Software Learn Solutions Docs

Search Contact Sign In Try Free

## MongoDB Atlas

Move faster with a cloud MongoDB service. Built for agile teams who'd rather spend time building apps than managing databases. Available on AWS, Azure, and GCP.

Start free

Already have an account? [Log in here](#) →

### Cloud Provider & Region

Choose your preferred cloud provider and the region nearest to clients

Select a cloud provider to see its region availability

aws google cloud azure

Configure a free tier cluster by first selecting a region labeled with [FREE TIER AVAILABLE](#). Then choose the M0 option in the Cluster Tier below.

recommended region (3)

NORTH AMERICA	EUROPE	ASIA
<ul style="list-style-type: none"><li>N. Virginia (us-east-1) <a href="#">FREE TIER AVAILABLE</a></li><li>Ohio (us-west-1)</li><li>N. California (us-west-1)</li><li>Oregon (us-west-2)</li><li>Montreal (ca-central-1)</li></ul>	<ul style="list-style-type: none"><li>Ireland (eu-west-1) <a href="#">FREE TIER AVAILABLE</a></li><li>London (eu-west-2)</li><li>Frankfurt (eu-central-1) <a href="#">FREE TIER AVAILABLE</a></li><li>Stockholm (eu-north-1)</li><li>São Paulo (sa-east-1)</li></ul>	<ul style="list-style-type: none"><li>Tokyo (ap-northeast-1)</li><li>Bangkok (ap-southeast-1)</li><li>Singapore (ap-southeast-1)</li><li>Mumbai (ap-south-1)</li></ul>

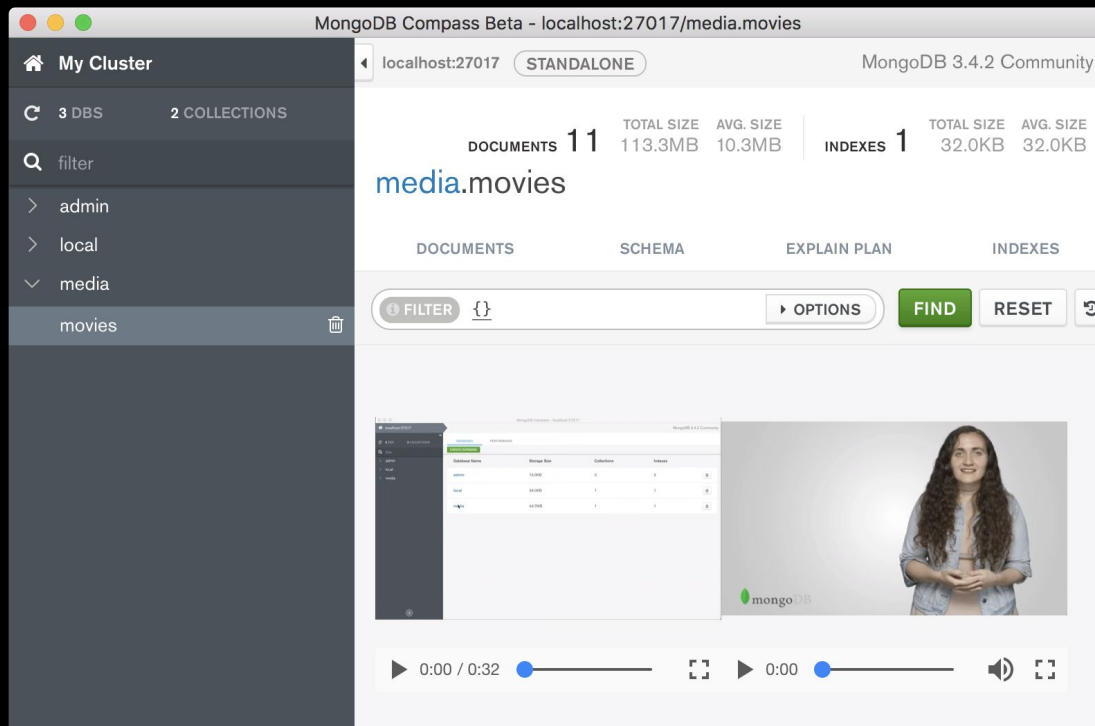
Pricing Getting started Migrate to MongoDB Atlas Frequently Asked Questions

MongoDB Atlas is the global cloud database service for modern applications. Deploy fully managed MongoDB across AWS, Azure, or GCP. Best-in-class

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

# mongodb

# compass



<https://www.mongodb.com/products/compass>



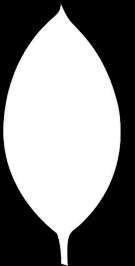
# Crud

```
bash

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

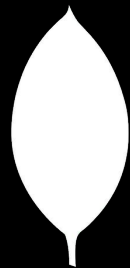
$
```

**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)  
})  
  
...
```





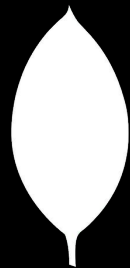
index.js

...

```
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})  
    }  
  }  
}
```

...

# find



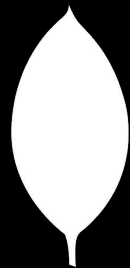
index.js

```
...
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})
    }
  }
}
...

```

# Find one



index.js

update

...

```
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

update

```
...
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**



```
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 **compass & crud**

# Sessions

# sessions

compass

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

**express-session** handles  
sessions through cookies

```
bash
$ npm install express-session
+ express-session@1.15.6
added 5 packages in 1.749s
$
```

index.js

```
...  
var session = require('express-session')  
  
...  
  
express()  
  ...  
  .use(session({  
    resave: false,  
    saveUninitialized: true,  
    secret: process.env.SESSION_SECRET  
  })))  
  ...  
  .listen(8000)
```



Developer Tools - http://localhost:8000/

Elements Console Sources **Network** Performance Memory Application Security Audits ChromeLens aXe

View: [Icons] Group by frame [ ] Preserve log [x] Disable cache [ ] Offline Online [v]

Filter [ ] Hide data URLs [All] XHR JS CSS Img Media Font Doc WS Manifest Other

20 ms 40 ms 60 ms 80 ms 100 ms 120 ms 140 ms 160 ms 180 ms 200 ms

Name [x] Headers Preview Response Cookies Timing

index.css  
index.js  
**localhost**

**General**

Request URL: http://localhost:8000/  
Request Method: GET  
Status Code: 200 OK  
Remote Address: [::1]:8000  
Referrer Policy: no-referrer-when-downgrade

**Response Headers** view source

Connection: keep-alive  
Content-Length: 691  
Content-Type: text/html; charset=utf-8  
Date: Sat, 03 Mar 2018 16:34:24 GMT  
ETag: W/"2b3-AFSSiIE4iiJ/cIAFFetX3K5zPxg"  
set-cookie: connect.sid=s%3A6CgnaCco\_jYNmLQ-plPaLrn3mUocEqLV.NyvXixDaLI6RdkS9ogo6vVGntocjaKzRztv%2BAYxZ74I; Path=/; HttpOnly  
X-Powered-By: Express

**Request Headers** view source

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8  
Accept-Encoding: gzip, deflate, br  
Accept-Language: en-GB,en;q=0.9,en-US;q=0.8,nl;q=0.7  
Cache-Control: no-cache  
Connection: keep-alive  
DNT: 1  
Host: localhost:8000  
Pragma: no-cache  
Referer: http://localhost:8000/  
Upgrade-Insecure-Requests: 1  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_13\_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.186 Safari/537.36

3 requests | 2.5 KB transferred | F...

index.js

...

```
function form(req, res) {  
  if (req.session.user) {  
    res.render('add.ejs')  
  } else {  
    res.status(401).send('Credentials required')  
  }  
}
```

...

## view/list.ejs

```
<% include head.ejs %>
<title>Movies - My movie website</title>
<h1>Movies</h1>
<p>
  <% if (user) { %>
    Hello <%= user.username %>!
  <% } else { %>
    <a href=/log-in>Log in</a>
    or
    <a href=/sign-up>Sign up</a>
  <% } %>
</p>

...

<% if (user) { %>
  <a href=/add>Add a movie</a>
<% } %>
<% include tail.ejs %>
```

# storage



Store user input in a MongoDB database instead of the server.

### Synopsis

- **Time:** 8:00h
- **Goals:** subgoal 7, subgoal 8
- **Due:** before week 5

### Description

A database will permanently store the data for you. A database is for persistence, store information apart from the server. Text is the easiest thing to store, media such as images, blobs or passwords require additional effort.

work on **storage**

&lt;&gt; Code

Issues 37

Pull requests 0

Actions

Projects 1

Security

Insights

Settings

Branch: master

be-course-19-20 / assessments / a1.md

Find file

Copy path



dandevri Update a2 rubric

39ac062 on Jan 16

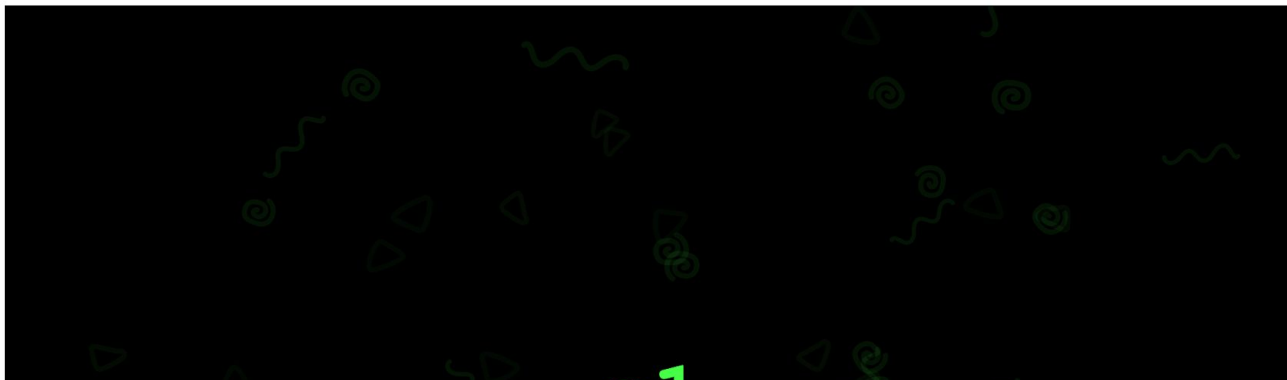
1 contributor

77 lines (63 sloc) | 4.5 KB

Raw

Blame

History



have a look at the **rubric for A1**



dandevri Update a2 rubric

39ac062 on Jan 16

1 contributor

77 lines (63 sloc) 4.5 KB

Raw

Blame

History



have a look at the **peer review document**



# exit;

see you in lab-4b!