Unit 12: MongoDB.

2021/2022

Free courses!!

- M001: MongoDB Basics
- M042: New Features and Tools in MongoDB 4.2
- M103: Basic Cluster Administration
- M121: The MongoDB Aggregation Framework
- M201: MongoDB Performance
- M220J: MongoDB for Java Developers
- M220JS: MongoDB for Javascript Developers
- And many more: https://university.mongodb.com/courses/catalog

Contents

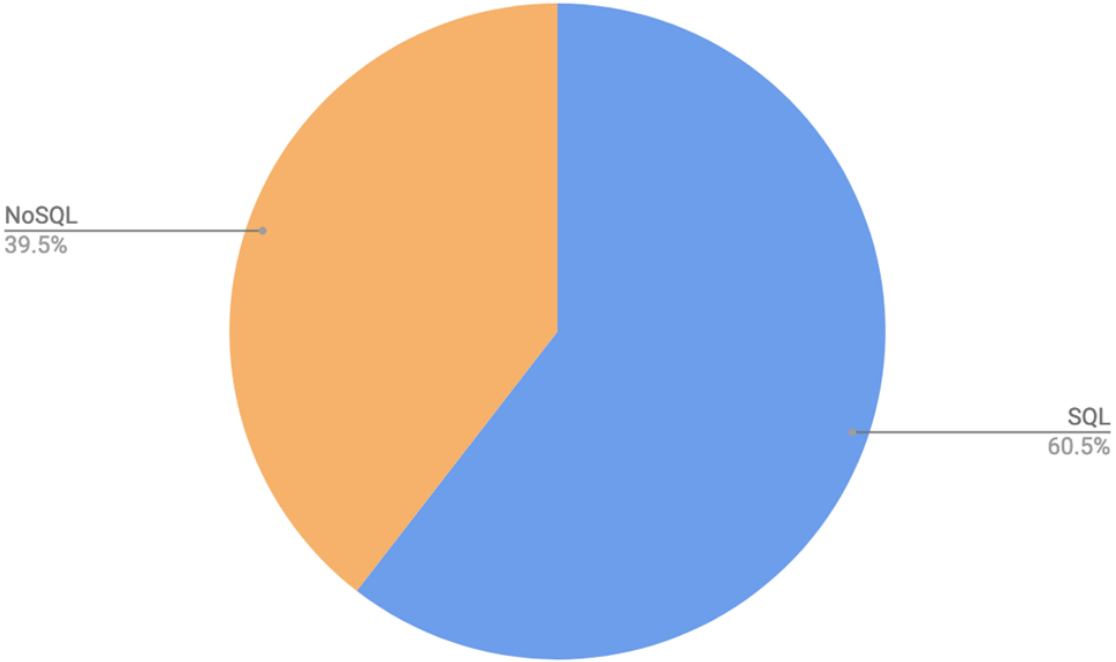
12.1. Introduction.

12.1.1. What is MongoDB?

Source: Wikipedia

- **MongoDB is a** cross-platform **document-oriented database management system**. In a nutshell: **it is a database solution**.
- English word: huMONGOus.
- **Classified as a NoSQL DBMS, MongoDB uses JSON-like documents.**
- MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL).
- Website: https://www.mongodb.com/

12.1.1. What is MongoDB?

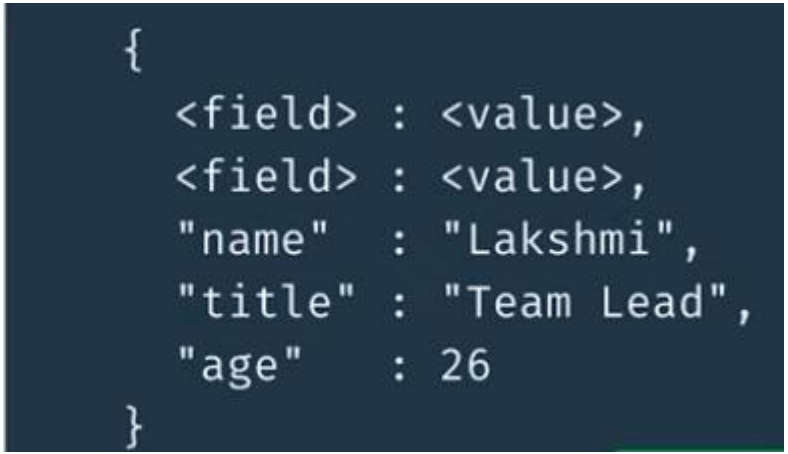


[2019 Database Trends – SQL vs. NoSQL, Top Databases, Single vs. Multiple Database Use]

12.1.1. What is MongoDB?

Document

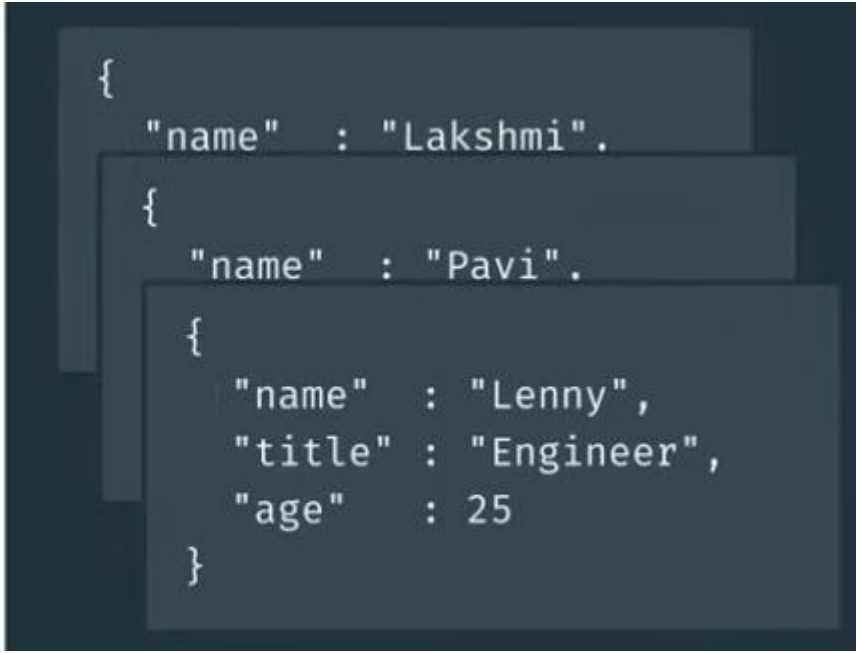
A way to organize and store data as a set of field-value pairs



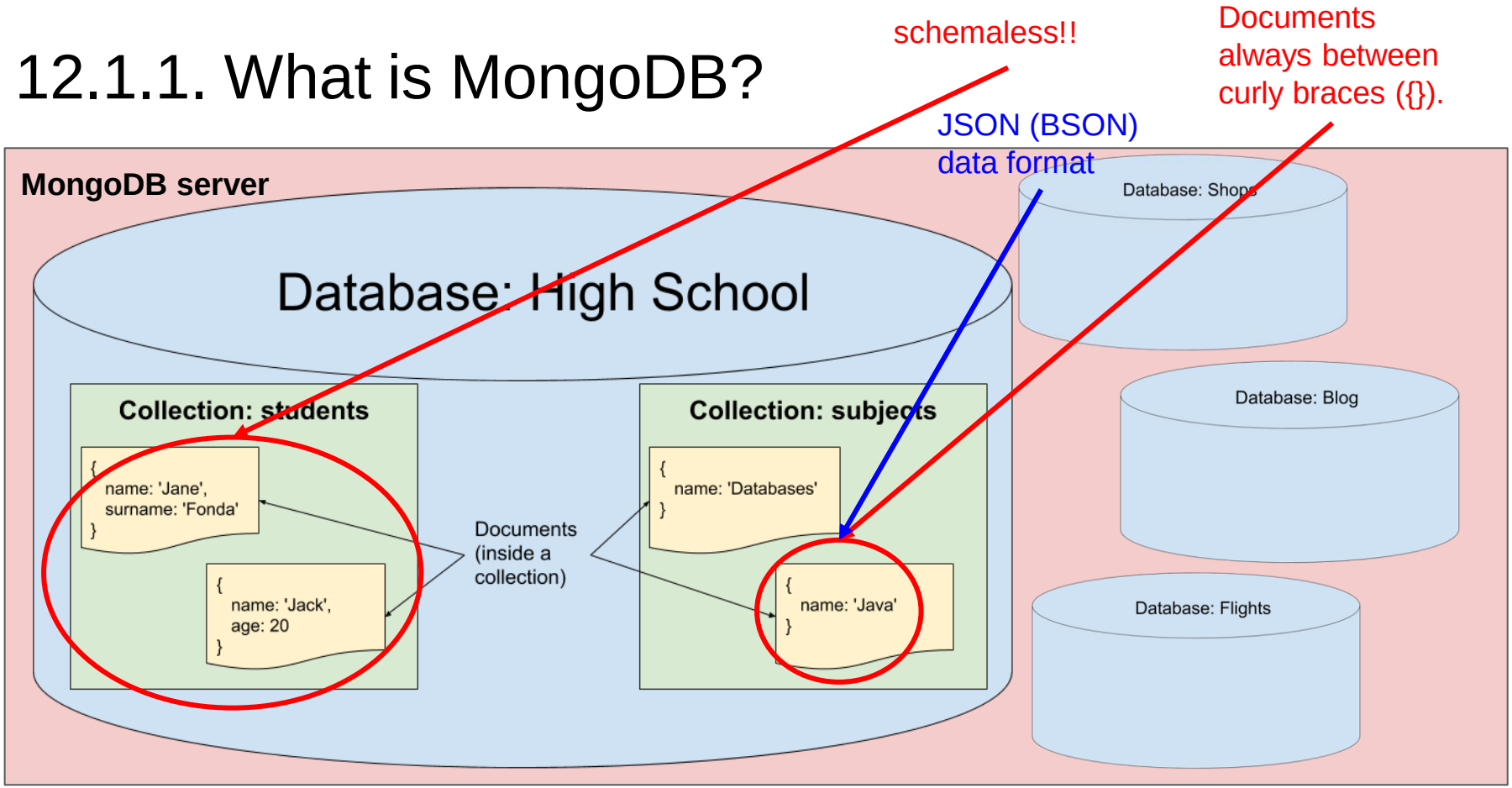
12.1.1. What is MongoDB?

Collection

An organized store of documents in MongoDB, usually with common fields between documents



12.1.1. What is MongoDB?



12.1.1. What is MongoDB?

JavaScript Object Notation

- JSON (BSON) data format:

Field = key + value

Key: Value:

```
{
    "name": "Sergi",
    "surname": "González",
    "address":
        {
            "street": "Caracas",
            "city": "Palma"
        },
    "interests": [
                { "name": "Databases"
},
                { "name":
"Javascript" }
    ]
}
```

JSON is converted internally to a Binary version of JSON

Document inside a collection of a database

Nested data (way to create relationships between data)

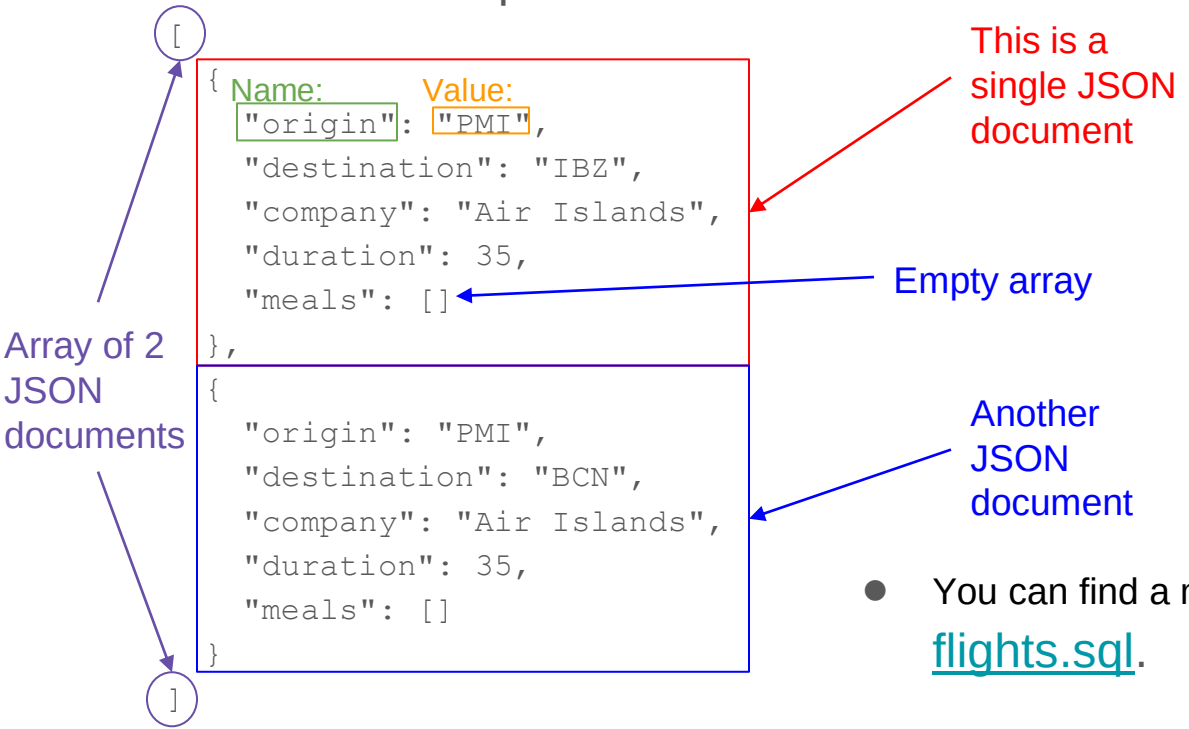
Embedded documents inside an array

12.1.1. What is MongoDB?

Mapping Terms and Concepts from SQL to MongoDB

12.1.2. BSON Data Structure.

- JSON file example:

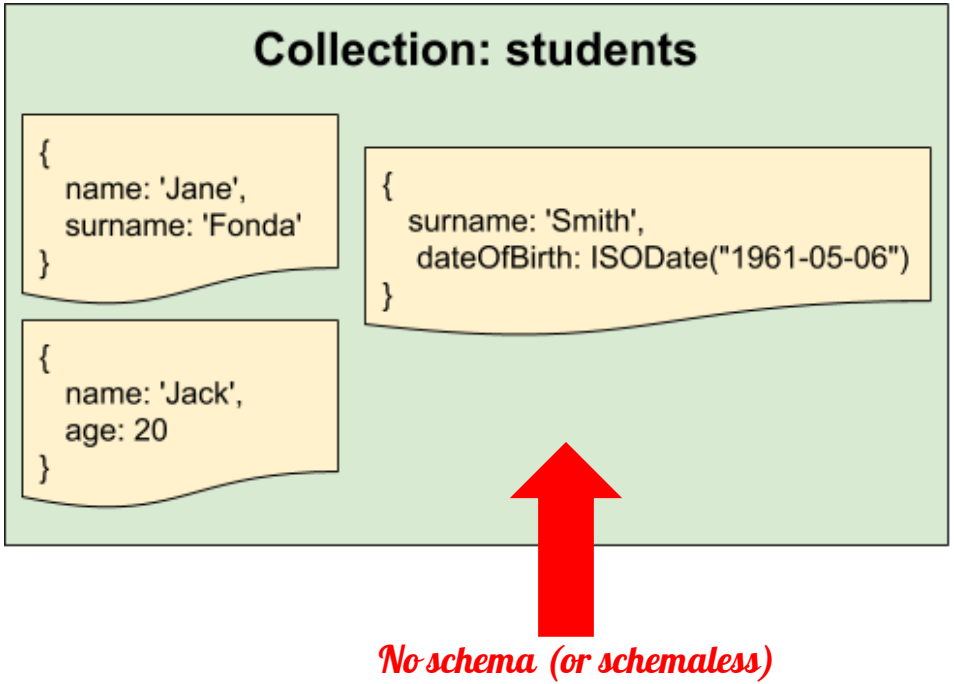


- Watch: https://www.youtube.com/watch?v=iiADhChRriM

- You can find a more complex example here flights.sql.

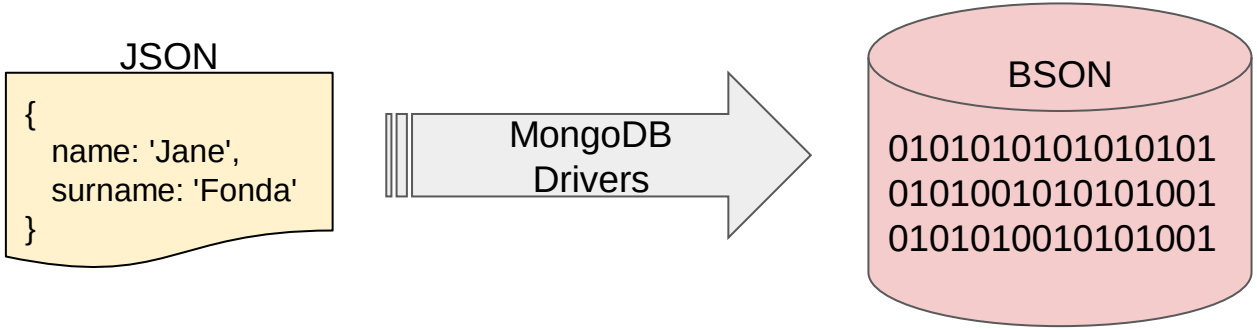
12.1.2. BSON Data Structure.

- MongoDB is NoSQL, that is:
  - We do not normalize data (sometimes we store data together in a JSON document).
  - We may not have a schema in a MongoDB database (documents may have different structure inside the same collection).
- Structured data (e.g. PostgreSQL) VS Flexibility (MongoDB).
- Relational model needs to JOIN relations to access related data. With MongoDB you may have related data inside the same document (or maybe not!!).



12.1.2. BSON Data Structure.

- **JSON is stored internally as BSON for efficiency**.
  - BSON: Binary data.
  - BSON: extends JSON Types (e.g. more detailed data types for numbers).
  - BSON: Efficient storage
  - BSON specification: https://www.json.org/ and http://bsonspec.org/
  - Cool tool: https://next.json-generator.com/

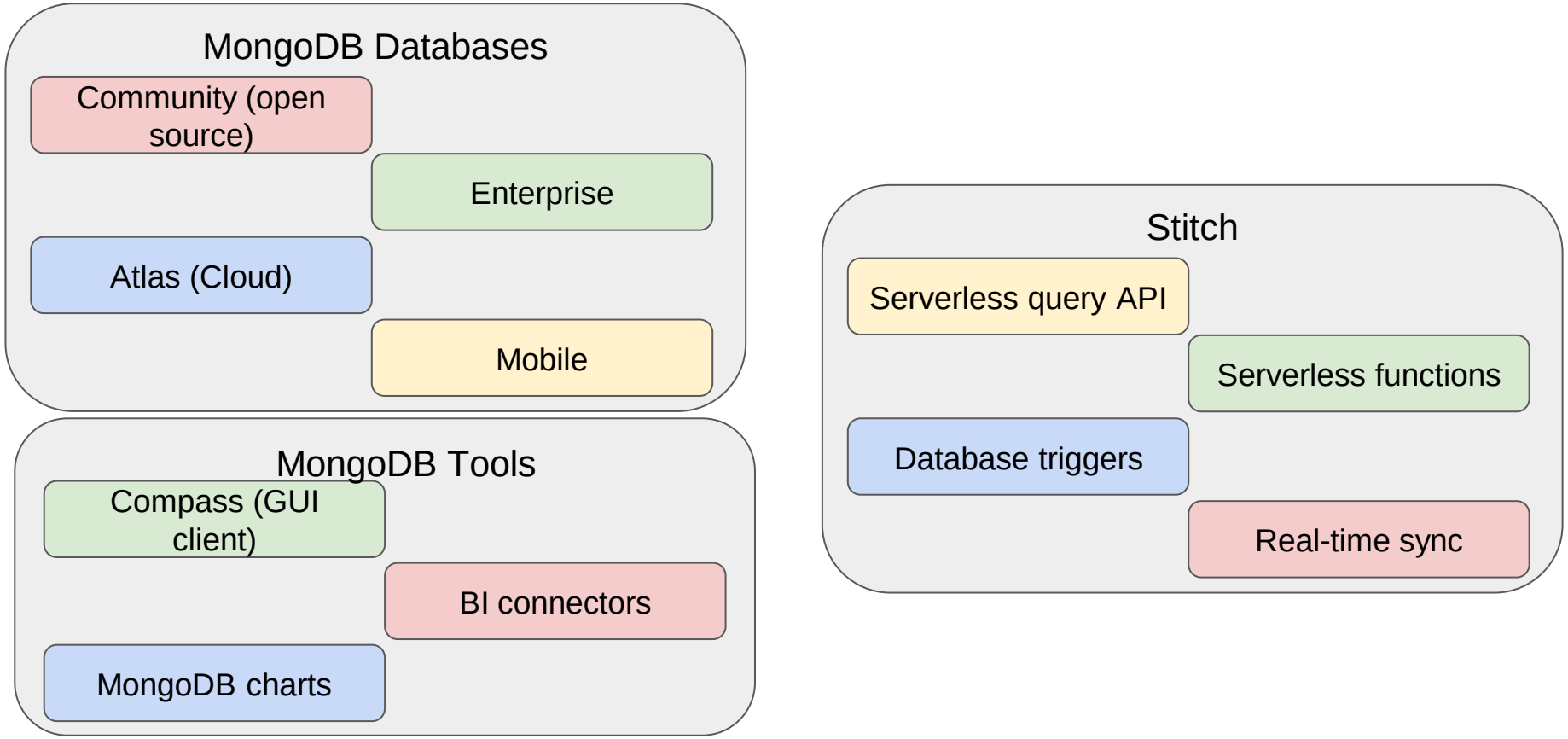


https://www.mongodb.com/json-and-bson

12.1.3. MongoDB ecosystem.

- MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL).
- Server Side Public License FAQ: https://www.mongodb.com/licensing/server-side-public-license/faq

12.1.3.MongoDB ecosystem.



12.1.3. MongoDB ecosystem.

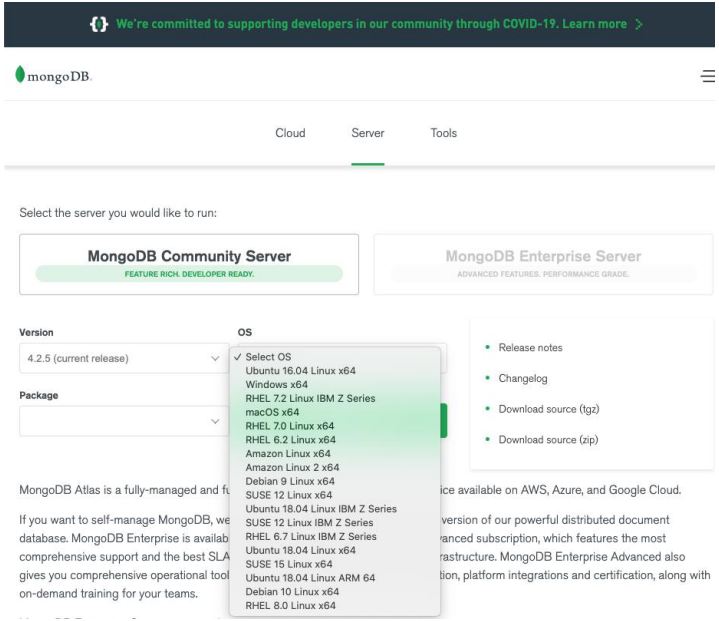
- "**MongoDB Database**: This is obvious, the database you want to work on it. It has community edition and has all the features you need for now.
- **Atlas (Cloud)**: You can focus your data and all the system and ops related tasks done for you.
- **Mobile**: MongoDB Mobile lets you store data where you need it, from iOS, Android, and IoT devices to your backend in the cloud – all using a single database. MongoDB Mobile lets you build the fastest, most reactive, always-on apps.
- **Compass**: This is the user interface to manage your MongoDB. Visually explore your data. Run ad hoc queries in seconds. Interact with your data with full CRUD functionality. View and optimize your query performance.
- **BI Connectors**: The MongoDB BI Connector lets you use MongoDB as a data source for your BI and analytics platforms. Seamlessly create the visualizations and dashboards that will help you extract the insights and hidden value in your multi-structured data.
- **MongoDB Charts**: [...] MongoDB Charts is the fastest and easiest way to create visualizations of MongoDB data. Connect to any MongoDB instance as a data source, create charts and graphs, build dashboards, and share them with other users for collaboration."

Source: mrtgr.com

- "**Stitch**:
  - Exposes the full power of working with documents in MongoDB and the MongoDB query language, directly from your web and mobile application front-end code. A powerful rules engine lets developers declare fine-grained security policies.
  - Allows developers to run simple JavaScript functions in the Stitch serverless platform, making it easy to implement application logic, securely integrate with cloud services and microservices, and build APIs.
  - Executes functions in real time in response to changes in the database or user authentication events.
  - Automatically synchronizes data between documents held locally in MongoDB Mobile and the backend database.
  - This is the snapshot of the ecosystem. You can find much more detail on http://www.mongodb.com . I'm very excited about Stitch and its abilities. See you in the next article."

12.1.4. Installing MongoDB.

- Available for MacOS, Windows and Linux.
- Community Server is the free edition, download it from here: https://www.mongodb.com/download-center/community



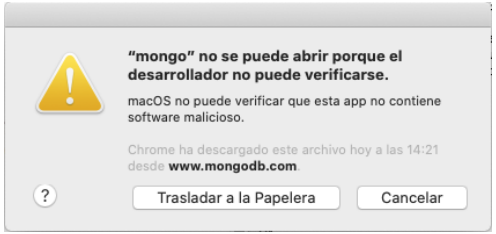
12.1.4. Installing MongoDB.

Source: MongoDB

- Official installation guides:
  - Windows: https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/
  - macOS: https://docs.mongodb.com/manual/tutorial/install-mongodb-on-os-x/
  - Linux: https://docs.mongodb.com/manual/administration/install-on-linux/

Let's practice

- Install mongodb in your computer following the video that we watched before and reading the official installation guide for your operating system.
- With newer versions of MacOS you may get this problem running mongo and mongod:



- To solve it just run one time the program from Finder:

Let's practice

linux shell → $
mongo shell → >

- After all the installation process you will be able to access mongodb from mongo shell:

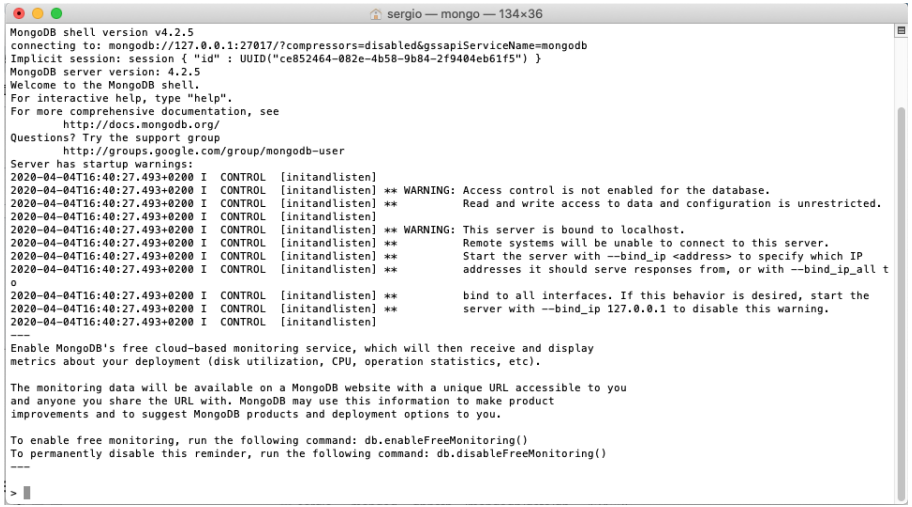
default port 27017

MongoDB server running:

```
$mongod --dbpath /Users/sergio/mongodb/data/db
$mongod --dbpath /Users/sergio/mongodb/data/db --port 27018
```

Accessing MongoDB server from mongo shell:

```
$mongo
$mongo --port 27018
```



Help for MongoDB server: $mongod --help
Help for MongoDB client: $mongo --help -> inside the client: > help

12.1.4. Installing MongoDB.

Source: MongoDB

- If you want to run MongoDB setting a log file:
  - First create the directory if it doesn't exist:
    - `mkdir /Users/sergi/mongodb/logs`
  - `mongod --dbpath /Users/sergi/mongodb/data/db --logpath /Users/sergi/mongodb/logs/log.log`
- If you want to run MongoDB as a background process `--fork`:
  - `mongod --fork --dbpath /Users/sergi/mongodb/data/db --logpath /Users/sergi/mongodb/logs/log.log`
- If you want to shutdown the server to which you are connected from "`mongo`" client:
  - `use admin`
  - `db.shutdownServer()`

12.1.4. Installing MongoDB.

Source: MongoDB

- You can save your setting in a file named “mondo.cfg”:
  - `vi /Users/sergio/mongodb/bin/mongod.cfg`

```
storage:
        "/Users/sergio/mongodb/data/db"
systemLog:
        destination: file
        path: "/Users/sergio/mongodb/logs/log.log"
```

  - To run MongoDB using this file:
    - `sudo mongod -f /Users/sergio/mongodb/bin/mongod.cfg`
- You have more examples here:

```
systemLog:
   destination: file
   path: "/var/log/mongodb/mongod.log"
   logAppend: true
storage:
   journal:
      enabled: true
processManagement:
   fork: true
net:
   bindIp: 127.0.0.1
   port: 27017
setParameter:
   enableLocalhostAuthBypass: false
...
```

12.1.4. Installing MongoDB.

- Configuration File Options: https://docs.mongodb.com/manual/reference/configuration-options/
- mongo (shell client): https://docs.mongodb.com/manual/reference/program/mongo/
- mongod (daemon process): https://docs.mongodb.com/manual/reference/program/mongod/

12.1.5. Getting Started.

> x = 200;

> x / 5;

> function factorial (n) {if (n <= 1) return 1; return n * factorial(n – 1);}

> factorial(5);

```
> x = 200;
200
> x / 5;
40
```

```
> function factorial (n) {
... if (n <= 1) return 1;
... return n * factorial(n - 1);
... }
> factorial(5);
120
```

12.1.5. Getting started.

- Show existing databases:
  - `> show dbs`
- Connect to a database:
  - `> use shop`

- Show data in a collection (pretty):
  - `> db.products.find().pretty()`

```
{
        "_id" :
ObjectId("5e89d4f1dd878b7131f6aa93"),
        "name" : "Little Prince",
        "price" : 12.99
}
```

You can set "_id" but it must be unique (concept of PK in relational model).

- Create a collection:
  - `> db.products.insertOne({name: "Little Prince", price: 12.99})`

```
{
        "acknowledged" : true,
        "insertedId" :
ObjectId("5e89d4f1dd878b7131f6aa93")
}
```

MongoDB automatically generates a unique ID

- Show data in a collection:
  - `> db.products.find()`

```
{ "_id" : ObjectId("5e89d4f1dd878b7131f6aa93"),
"name" : "Little Prince", "price" : 12.99 }
```

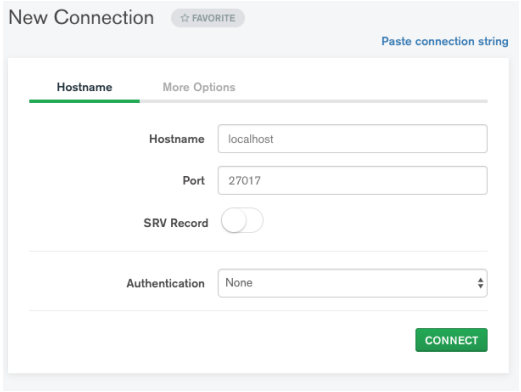
- Show collections:
  - `> show collections`

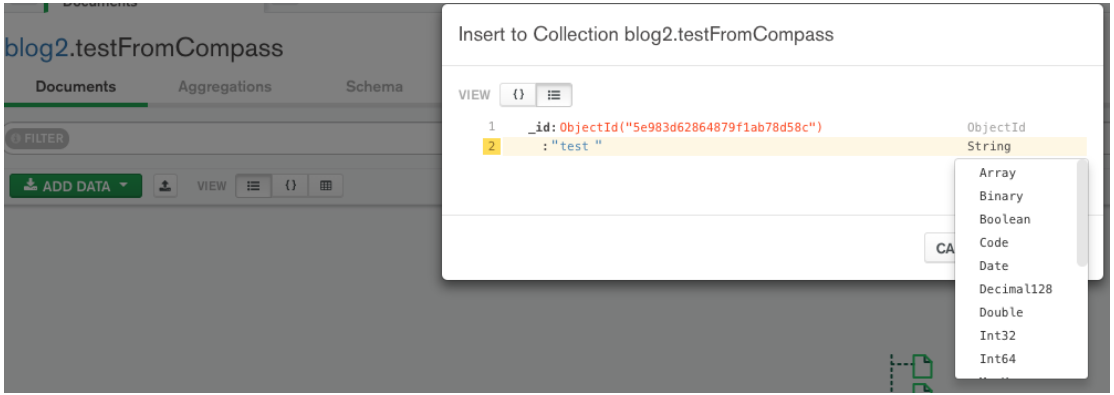
12.1.5. Getting started.

- To remove a database:
  - `use databaseName`
  - `db.dropDatabase()`
- To remove a collection:
  - `db.myCollection.drop()`
- To get statistics about a database:
  - `db.stats()`
- Current DB:
  - `db.getName() (or simply "db")`
- Clean the screen:
  - `cls`

12.1.5. Getting started.

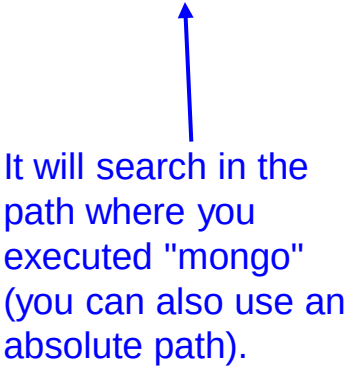
- As you know, I'll use only the shell. But there is a visual tool (now free for all!): MongoDB Compass.
- It's up to you to use it. A good video about Compass (Spanish): https://youtu.be/dBpRBz5ye-g





12.1.5. Getting started.

- You can load data with load() function.
  - > load(file.js)



file.js

```
db = db.getSiblingDB("myDB");
db.myCol.drop();
db.myCol.insertMany([
…
]);
```

12.1.6. Shell vs drivers.

Source: Academind

- Drivers: Drivers are “bridges” between your programming language and MongoDB server.
- Drivers documentation; https://docs.mongodb.com/drivers/

Java:

```
import com.mongodb.ConnectionString;
import com.mongodb.client.MongoClients;
import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoDatabase;
import com.mongodb.MongoClientSettings;

// ...
ConnectionString connString = new ConnectionString(
    "mongodb+srv://<username>:<password>@<cluster-address>/test?w=majority"
);
MongoClientSettings settings = MongoClientSettings.builder()
    .applyConnectionString(connString)
    .retryWrites(true)
    .build();
MongoClient mongoClient = MongoClients.create(settings);
MongoDatabase database = mongoClient.getDatabase("test");
```

Python:

```
import pymongo

client = pymongo.MongoClient(
   "mongodb+srv://<username>:<password>@<cluster-url>/test?retryWrites=true&w=majorit
y")
db = client.test
```

Do you understand the difference between working with the shell and using a driver to connect from a programming language?

12.1.6. Shell vs drivers.

```
$ python -m pip install pymongo
$ python -m pip install PyHamcrest
$ python
>>> from pymongo import MongoClient
>>> from pprint import pprint
>>> client = MongoClient("localhost")
>>> db = client.contactsDB
>>> contact = db.contacts.find_one({})
>>> print(contact)
```

Source: https://dzone.com/articles/getting-started-with-python-and-mongodb

12.1.6. Shell vs drivers.

```
>>> contacts = db.contacts.find({})
>>> for doc in contacts:
>>>     print(doc)
```

Source: https://dzone.com/articles/getting-started-with-python-and-mongodb

12.1.7. Basic architecture.