Tecnologie e applicazioni web

MongoDB

Filippo Bergamasco (filippo.bergamasco@unive.it)
http://www.dais.unive.it/~bergamasco/
DAIS - Università Ca'Foscari di Venezia
Academic year: 2023/2024

About MongoDB

MongoDB is a DBMS:

- Non-relational
- Oriented to documents (and not to data relations)
- With a dynamic schema (schema-less)
- JSON-style documents

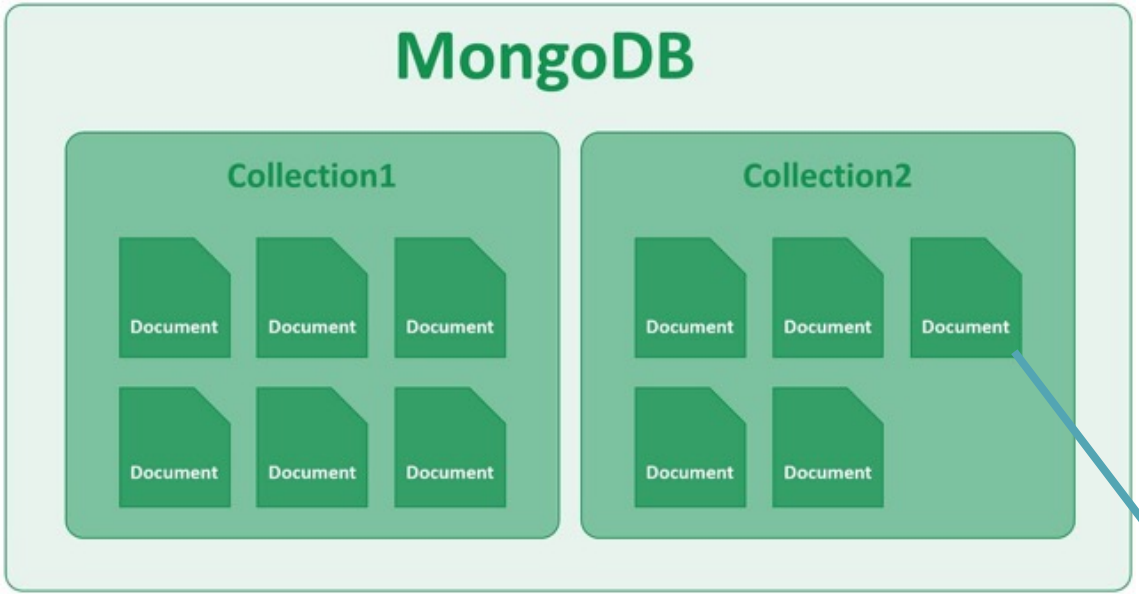
Why is it interesting?
Excellent integration with dynamic languages like JavaScript. Database structure can change on the fly while developing our application

MongoDB

https://docs.mongodb.com/manual/#

The DBMS allows the creation of multiple **databases**.

Each **database** is composed of **collections**
Each **collection** is a set of **documents**
Each **document** is composed of one or more **fields**



fields

```
{
  "student": {
    "name": "John",
    "class": "Intermediate",
    "address": {
      "street": "2293 Example Street",
      "City": "Chicago",
      "State": "IL"
    }
  }
}
```

Terminology

RDBMS		MongoDB
Database	→	Database
Table	→	Collection
Index	→	Index
Row	→	Document
Column	→	Field
Join	→	Embedding & Linking

MongoDB vs. relational

- Each document in a collection can be composed of a different set of different fields
  - Increased flexibility since data can be stored and loaded without a predefined schema
- Each document can contain other documents (Embedding)
  - This mechanism can replace the usual join operation in relational databases.

One-to-many relations

Two alternatives:

1. By embedding documents inside the same parent document (fast readings but might be more complex to keep data consistency)
2. By referencing document ids like in relational databases (slower readings, but data is not replicated)

Embedding vs. referencing

```
{
   "_id": "joe",
   "name": "Joe Bookreader",
   "addresses": [
      {
         "street": "123 Fake Street",
         "city": "Faketon",
         "state": "MA",
         "zip": "12345"
      },
      {
         "street": "1 Some Other Street",
         "city": "Boston",
         "state": "MA",
         "zip": "12345"
      }
   ]
}
```

```
{
   _id: "oreilly",
   name: "O'Reilly Media",
   founded: 1980,
   location: "CA"
}

{
   _id: 123456789,
   title: "MongoDB: The Definitive Guide",
   author: [ "Kristina Chodorow", "Mike Dirolf" ],
   published_date: ISODate("2010-09-24"),
   pages: 216,
   language: "English",
   publisher_id: "oreilly"
}

{
   _id: 234567890,
   title: "50 Tips and Tricks for MongoDB Developer",
   author: "Kristina Chodorow",
   published_date: ISODate("2011-05-06"),
   pages: 68,
   language: "English",
   publisher_id: "oreilly"
}
```

Mongoose

Mongoose is a popular Object Document Mapping **ODM** library that maps JavaScript objects in MongoDB.

It allows us to define a document schema through JavaScript objects and to perform an automatic mapping from/to the database

http://mongoosejs.com/docs/guide.html

Mongoose

Mongoose's most important concepts are:

Schemas:

To describe the document structure of a collection (together with their methods!)

Models:

Are functions (constructors) to instantiate objects given a schema and store them automatically in the database

Mongoose models

Once a model is defined, it can be used to:

- Query the database
  - Ex: <model>.find({ })
- Create and store a new object
  - Ex: <model>.create({obj})
- Remove existing objects
  - Ex: <model>.remove({}) or <model>.deleteOne({})