JONAS.IO
SCHMEDTMANN

NODE.JS, EXPRESS & MONGODB
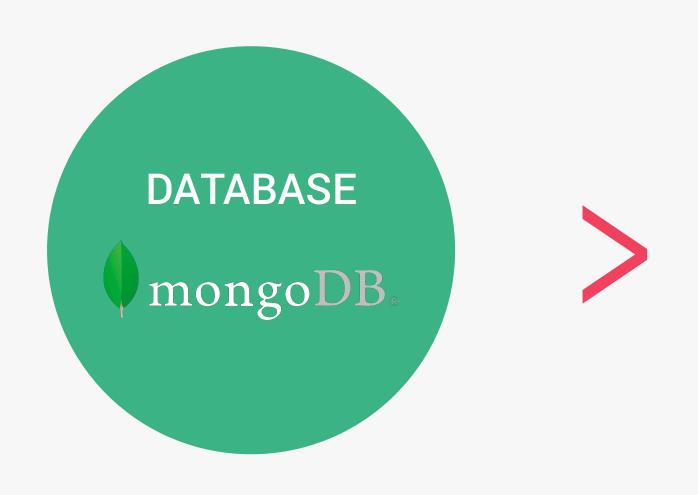
THE COMPLETE BOOTCAMP

@JONASSCHMEDTMAN

MONGODB: AN OVERVIEW

DATABASE

mongoDB.

DATABASE

👉 **NoSQL**

COLLECTIONS

("Tables")

blog	⟶
users	⟶
reviews	⟶

DOCUMENTS

("Rows")

| post |
| user |
| review |

MONGODB

"MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need"

KEY MONGODB FEATURES:

👉 **Document based:** MongoDB stores data in documents (field-value pair data structures, NoSQL);

👉 **Scalable:** Very easy to distribute data across multiple machines as your users and amount of data grows;

👉 **Flexible:** No document data schema required, so each document can have different number and type of fields;

👉 **Performant:** Embedded data models, indexing, sharding, flexible documents, native duplication, etc.

👉 Free and open-source, published under the SSPL License.

DOCUMENTS, BSON AND EMBEDDING

DOCUMENT STRUCTURE

👉 **BSON:** Data format MongoDB uses for data storage. Like JSON, **but typed**. So MongoDB documents are typed.

```
{
    "_id": ObjectID('9375209372634926'),
    "title": "Rockets, Cars and MongoDB",
    "author": "Elon Musk",
    "length": 3280,
    "published": true,
    "tags": ["MongoDB", "space", "ev"]
    "comments": [
        { "author": "Jonas", "text": "Interesting stuff!" },
        { "author": "Bill", "text": "How did oyu do it?" },
        { "author": "Jeff", "text": "My rockets are better" }
    ]
}
```
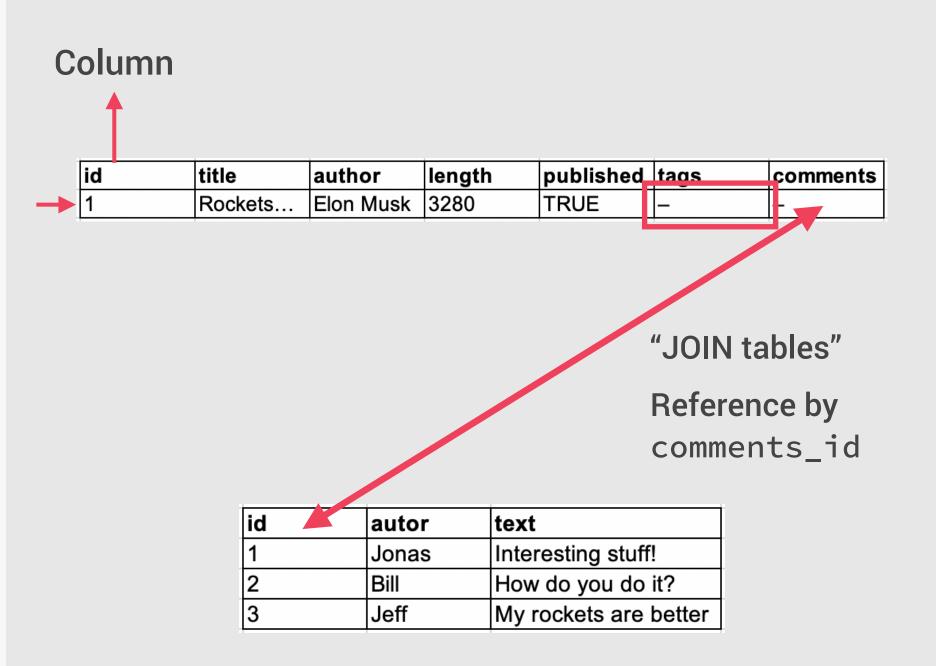
Unique ID

Fields

Values (*typed*)

Embedded documents

👉 **Embedding/Denormalizing:** Including related data into a single document. This allows for quicker access and easier data models (it's not always the best solution though).

RELATIONAL DATABASE

Column

id	title	author	length	published	tags	comments
1	Rockets…	Elon Musk	3280	TRUE	–	–

"JOIN tables"

Reference by `comments_id`

id	autor	text
1	Jonas	Interesting stuff!
2	Bill	How do you do it?
3	Jeff	My rockets are better

👉 **Data is always normalized**