Document stores

Introduction to NoSQL: Lecture 6

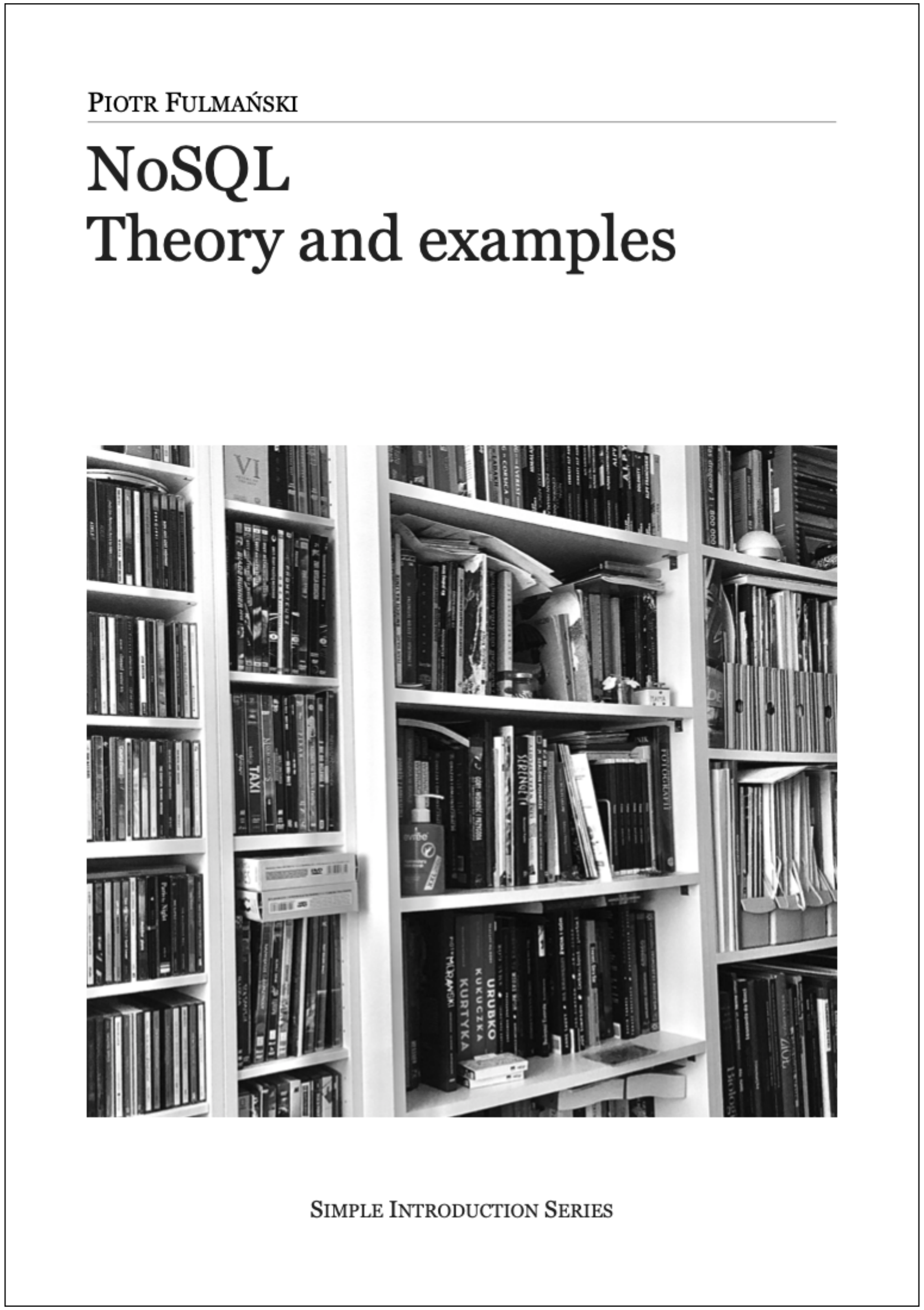
Piotr Fulmański

FACULTY OF MATHEMATICS
AND COMPUTER SCIENCE
University of Lodz

NoSQL Theory and examples

by Piotr Fulmański

Piotr Fulmański, 2021



Piotr Fulmański

NoSQL
Theory and examples

Simple Introduction Series

General overview of key-value stores

- Basic ideas and features
- Working with Apache CouchDB

Do we really need one more database type?

Do we really need one more database type?

- relational databases

Do we really need one more database type?

- relational databases
- column family stores

Do we really need one more database type?

- relational databases
- column family stores

- key-value stores

Do we really need one more database type?

- relational databases
- column family stores
- **document stores**
- key-value stores

Documents... What kind of documents?

Documents… What kind of documents?

Not `doc`, `xls` etc.

Documents... What kind of documents?

Document is a

Documents... What kind of documents?

Document is a ... **key-value pairs**.

Documents... What kind of documents?

Document is a **set** of ... **key-value pairs**.

```
{
  "customer": {
    "name": "Dart Vader",
    "location": "Star Destroyer"
  },
  "items": ["lightsaber", "black cloak", "air filter"]
}
```

can be expressed in the following XML form:

```
<document>
  <customer name="Dart Vader" location="Star Destroyer"/>
  <items>
    <item name="lightsaber"/>
    <item name="black cloak"/>
    <item name="air filter"/>
  </itmes>
</document>
```

Documents... What kind of documents?

Document is a **set** of **ordered** **key-value pairs**.

Documents... What kind of documents?

Document is a **set** of **ordered** **key-value pairs**.

We can think of a document store as a tree-like structure of objects being a collection of ordered key-value pairs.

Documents... What kind of documents?

Document is a **set** of **ordered** **key-value pairs**.

A matter of schemas existence

- Document stores do not require its structures to be formally defined or specified in any form.
- There is no formal definition of structure - in consequence we can faced with polymorphic schema.

Document stores vs. relational databases

- Normal forms. Need for data denormalization.
- Model determinism.
  - A relational database is driven primarily by the nature of the data to be stored, while a document store is driven by the nature of the queries to be executed.
  - In case of document stores considering database and models which are used, without telling about an application and the way data is going to be used, doesn't make any sense.
  - Data models used in document stores depends strictly on application needs.
- Too much abstraction is not advisable. Avoid too polymorphic schema.
- Schema-less can be tricky. Schema-less means freedom from the need to predefine the database structure ahead but never ever shouldn't be understand as a permission to mindless model expansion.
- More flexibility by a price of more responsibility

Summary

- Document stores are not about electronic documents like word processing, spreadsheet or any other businessman-readable file.
- We can think of a document store as a tree-like structure of objects being a collection of ordered key-value pairs.
- Although there are no constraints of a normal form type (known from relational model), we should be aware of not to fall into schema-less freedom trap when we loos a control over the type and variety of documents we store.
- We are more flexible when working with document stores compared to relational database, but in the same time have to be more responsible as the obligation to ensure data correctness and consistency has been transferred from the abstract theoretical model (what is a feature of relational model) to the user.

Bibliography

- [Ful] Piotr Fulmański, *NoSQL. Theory and examples*, Piotr Fulmański, 2021