AT82.02

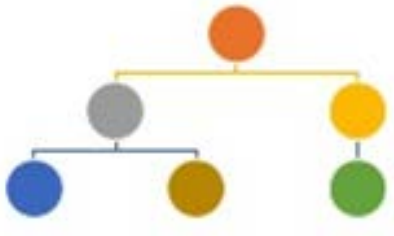
DATA MODELING AND MANAGEMENT

UNIT 2-2: DOCUMENT MODEL

CHUTIPORN ANUTARIYA (CHUTI AT AIT DOT AC DOT TH)

Co-funded by the Erasmus+ Programme of the European Union

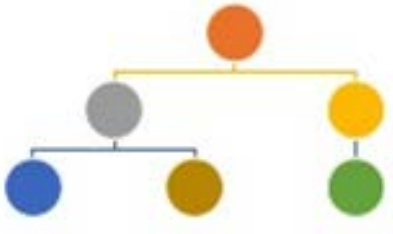
DS&AI



Document Model

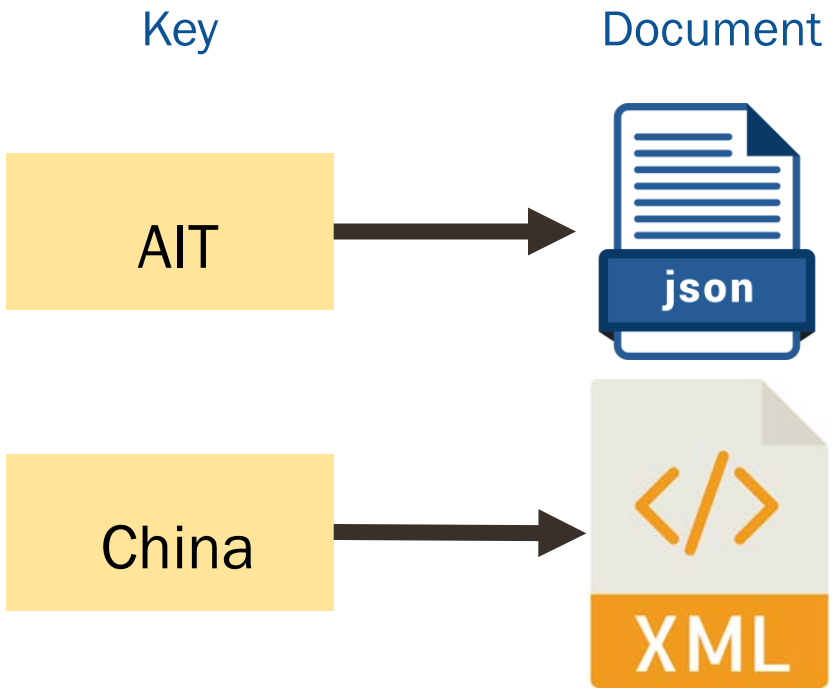
RECAP

Document Model

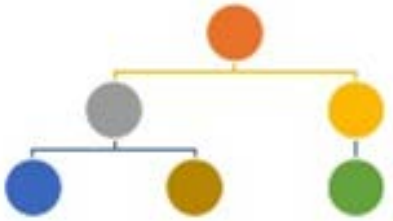


A specialized Key-value Store but rather than storing “values,” it stores “documents”, which are not adhered to schema restrictions.

Provides a way to query the documents based on the contents or metadata.



Document Model



A specialized Key-value Store

Designed for storing, retrieving and managing document-oriented information, also known as semi-structured data, such as XML, JSON, BSON

Provides APIs or a query/update language that exposes the ability to query or update based on the internal structure in the document.

RECAP

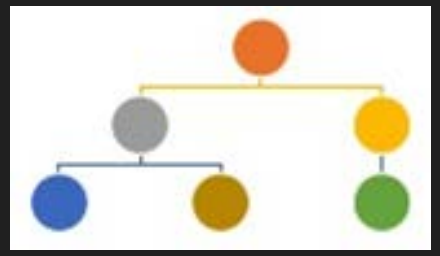
```
{
    "FirstName": "Bob",
    "Address": "5 Oak St.",
    "Hobby": "sailing"
}
```


```
<contact>
  <firstname>Bob</firstname>
  <lastname>Smith</lastname>
  <phone type="Cell">(123) 555-0178</phone>
  <phone type="Work">(890) 555-0133</phone>
  <address>
    <type>Home</type>
    <street1>123 Back St.</street1>
    <city>Boys</city>
    <state>AR</state>
    <zip>32225</zip>
    <country>US</country>
  </address>
</contact>
```

```
{
   "FirstName": "Bob",
   "Address": "5 Oak St.",
   "Hobby": "sailing"
}
```


```
<contact>
   <firstname>Bob</firstname>
   <lastname>Smith</lastname>
   <phone type="Cell">(123) 555-0178</phone>
   <phone type="Work">(890) 555-0133</phone>
   <address>
    <type>Home</type>
    <street1>123 Back St.</street1>
    <city>Boys</city>
    <state>AR</state>
    <zip>32225</zip>
    <country>US</country>
   </address>
  </contact>
```

CRUD Operations



- Creation (or insertion)
- Retrieval (or query, search, read or find)
- Update (or edit)
- Deletion (or removal)

About Document Model

Stores data in flexible, JSON-like documents

Fields can vary from document to document and data structure can be changed over time

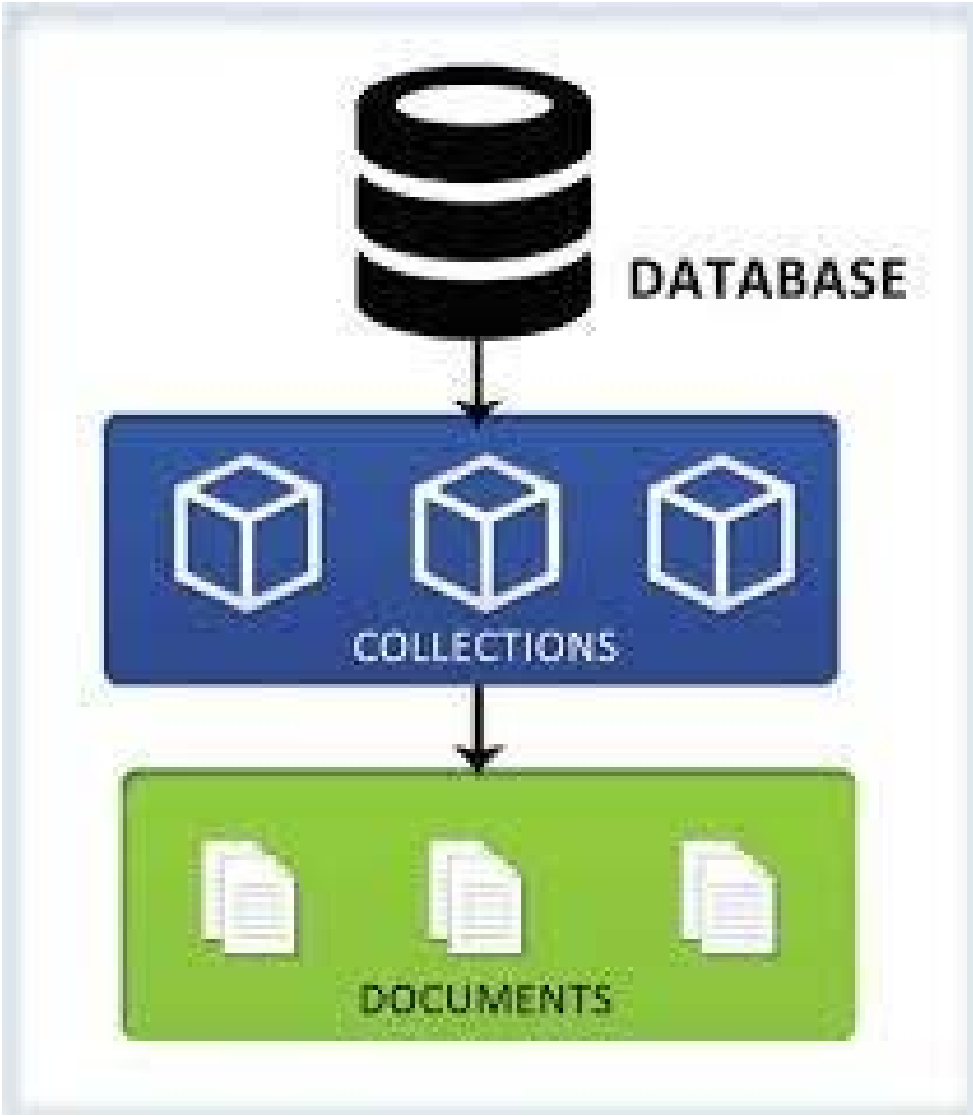
Queries, indexing, and aggregation

Example: MongoDB

Terminology

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

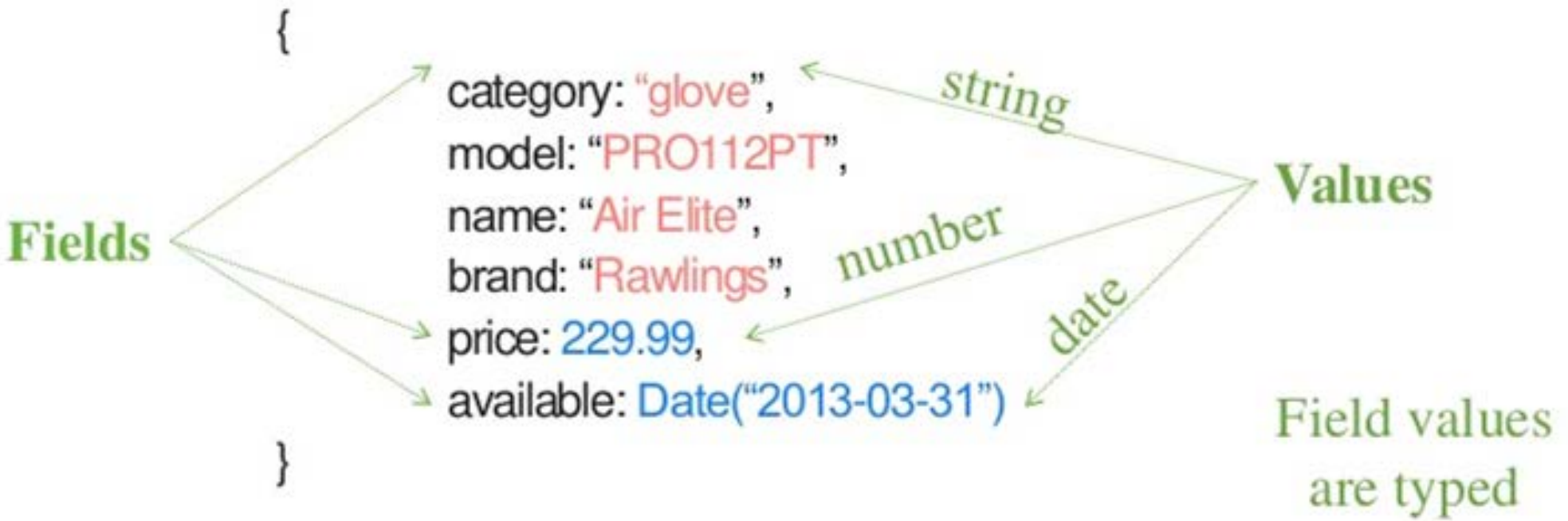
Database, Collections and Documents



How to model this in JSON?

Size: 12"
Infield/Outfield/Pitcher model
2-Piece Web pattern
Most popular MLB® pattern among pitchers
Pro Stock® American steerhide leather offers rugged durability and a superior feel
Dual-Welting™ on "exposed edges" of the fingers helps maintain pocket shape and durability
Pro Stock™ hand-designed pattern for unbeatable craftsmanship
Dri-Lex® ultra-breathable wrist lining repels moisture from your hand
Black leather with rich brown embellishments
Pattern: B212
Model: WTA2000BBB212
Wilson

JSON Object


{
category: "glove",
model: "PRO112PT",
name: "Air Elite",
brand: "Rawlings",
price: 229.99,
available: Date("2013-03-31")
}
Fields
Values
string
number
date
Field values
are typed

Documents are Rich Structures

```
{
        category: "glove",
        model: "PRO112PT",
        name: "Air Elite",
        brand: "Rawlings",
        price: 229.99,
        available: Date("2013-03-31"),
        position: ["infield", "outfield", "pitcher"]
}
```

Fields can contain arrays

Documents are Rich Structures

```
{
        category: "glove",
        model: "PRO112PT",
        name: "Air Elite",
        brand: "Rawlings",
        price: 229.99,
        available: Date("2013-03-31"),
        position: ["infield", "outfield", "pitcher"],
        endorsed: {name: "Ryan Howard",
                   team: "Phillies",
                   position: "first base"},
        history: [{date: Date("2013-03-31"),  price: 279.99},
                  {date: Date("2013-06-01"), price: 259.79},
                  {date: Date("2013-08-15"), price: 229.99}]
}
```

Fields can contain an array of sub-documents

Variation is Easy!

```
{

category: bat,
model: B1403E,
name: Air Elite,
brand: "Rip-IT",
price: 399.99

diameter: "2 5/8",
barrel: R2 Alloy,
handle: R2 Composite,
type: composite,

}
```

```
{

category: glove,
model: PRO112PT,
name: Air Elite,
brand: "Rawlings",
price: "229.99"

size: 11.25,
position: outfield,
pattern: "Pro taper",
material: leather,
color: black

}
```

```
{

category: ball,
model: ROML,
name: MLB,
brand: "Rawlings",
price: "6.99"

cover: leather,
core: cork,
color: white

}
```

Easy Query! (MongoDB Query)

```
> db.products.find( { "position" : "infield",
                      "endorsed.team" : "Phillies" } )

        {
                category: "glove",
                model: "PRO112PT",
                name: "Air Elite",
                brand: "Rawlings",
                price: 229.99,
                available: Date("2013-03-31"),
                position: ["infield", "outfield", "pitcher"],
                endorsed: {name: "Ryan Howard",
                           team: "Phillies",
                           position: "first base"},
        }
```

Object Relationships

1-1

Referencing & Embedding

https://docs.mongodb.com/manual/core/data-modeling-introduction/

1-1: General Recommendations

- **Embed**
  - No additional data duplication
  - Can query or index on embedded field
    - e.g., “result.type”
- Exceptional cases…
  - Embedding results in large documents
  - Set of infrequently access fields

```
{
    "_id": 333,
    "date": "2003-02-09T05:00:00",
    "hospital": "County Hills",
    "patient": "John Doe",
    "physician": "Stephen Smith",
    "type": "Chest X - ray",
    "result": {
        "type": "txt",
        "size": 12,
        "content": {
            "value1": 343,
            "value2": "abc"
        }
    }
}
```

1-M

1-M

Modeled in 2 possible ways

Embed

Patients

```
{
    _id: 2,
    first: "Joe",
    last: "Patient",
    addr: { ...},
    procedures: [
        {
            id: 12345,
            date: 2015-02-15,
            type: "Cat scan",
                ...},
        {
            id: 12346,
            date: 2015-02-15,
            type: "blood test",
                ...}]
}
```

Reference

Patients

```
{
    _id: 2,
    first: "Joe",
    last: "Patient",
    addr: { ...},
    procedures: [12345, 12346]}
```

Procedures

```
{
    _id: 12345,
    date: 2015-02-15,
    type: "Cat scan",
    ...}
{
    _id: 12346,
    date: 2015-02-15,
    type: "blood test",
    ...}
```

1-M : General Recommendations

- **Embed, when possible**
  - Many are weak entities
  - Access all information in a single query
  - Take advantage of update atomicity
  - No additional data duplication
  - Can query or index on any field
    - e.g., { “phones.type”: “mobile” }

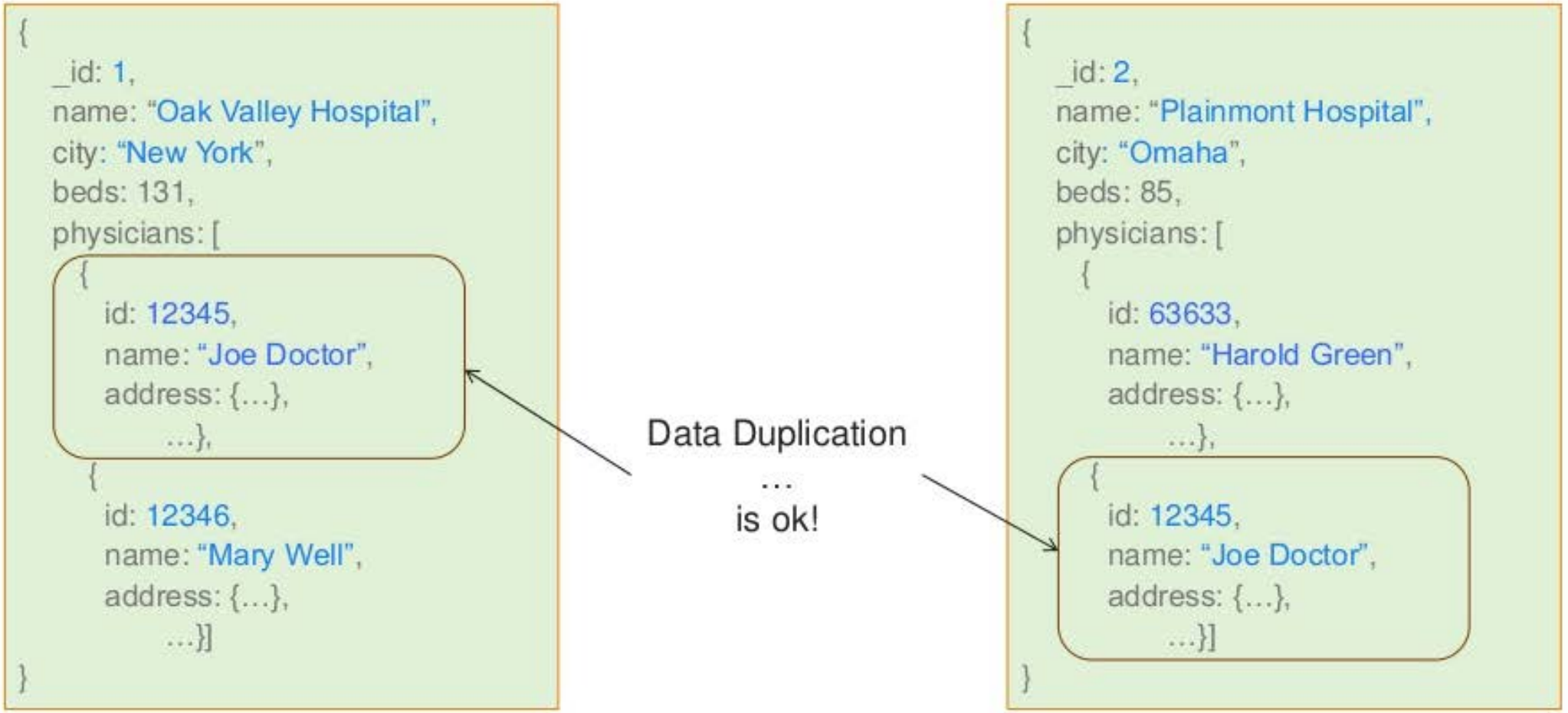
- Exceptional cases:
  - 16 MB document size
  - Large number of infrequently accessed fields

```
{
    _id: 2,
    first: “Joe”,
    last: “Patient”,
    addr: { …},
    procedures: [
      {
        id: 12345,
        date: 2015-02-15,
        type: “Cat scan”,
              …},
      {
        id: 12346,
        date: 2015-02-15,
        type: “blood test”,
              …}]
}
```

M–M

M-M

Embedding **Physicians** in **Hospitals** collection



M-M : General Recommendation

- **Use case determines whether to *reference* or *embed*:**
  1. Data Duplication
     - Embedding may result in data duplication
     - Duplication may be okay if reads dominate updates
       - Of the two, which one changes the least?
  2. Referencing may be required if many related items
  3. Hybrid approach
     - Potentially do both .. It's ok!

Hospitals

```
{
  _id: 2,
  name: "Oak Valley Hospital",
  city: "New York",
  beds: 131,
  physicians: [12345, 12346]}
```

Physicians

```
{
  _id: 12345,
  name: "Joe Doctor",
  address: {…},
  …}
{
  _id: 12346,
  name: "Mary Well",
  address: {…},
  …}
```

- P. Sadalage and M. Fowler: NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Addison-Wesley Professional, 2013
- Jan L. Harrington: Relational Database Design and Implementation, 4th edition, Morgan Kaufmann, 2016
- A. Makris, K. Tserpesa, V. Andronikou Dimosthenis Anagnostopoulos: A Classification of NoSQL Data Stores Based on Key Design Characteristics, Procedia Computer Science, Vol. 97, 2016, pp. 94-103.
- MongoDB Schema Design: Practical Applications and Implications [https://www.slideshare.net/mongodb/mongodb-schema-design-practical-applications-and-implications]