

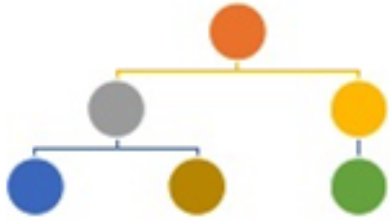
# AT82.02

DATA MODELING AND MANAGEMENT

---

UNIT 2-2: DOCUMENT MODEL

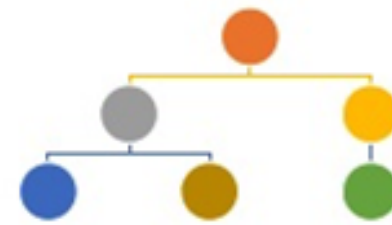
CHUTIPORN ANUTARIYA (CHUTI AT AIT DOT AC DOT TH)



# Document Model

---

# 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.

Key/ID/PK

Document

iPhone13

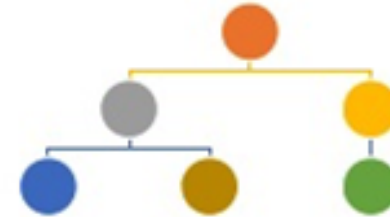


China





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

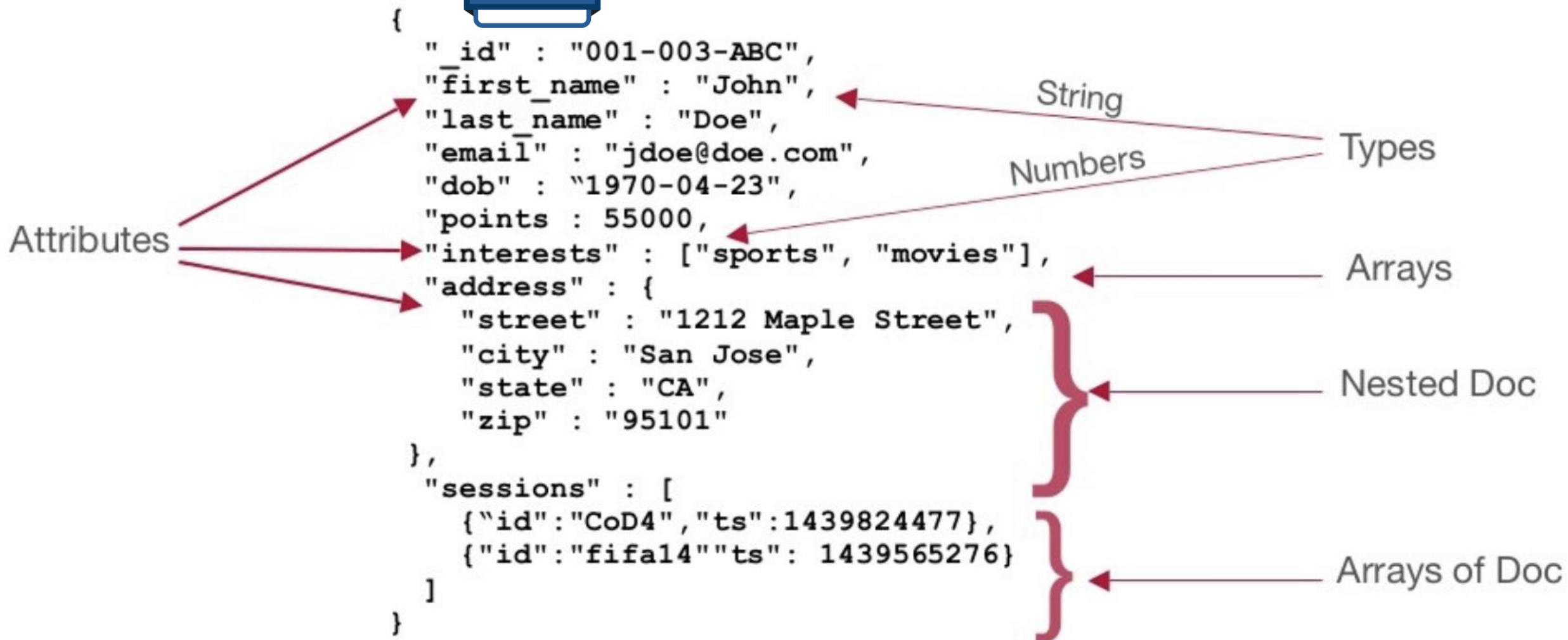
# Flexible Schema



- Support Data Types
- Complex Data Structure
- Developer Friendly
- Flexible

```
{
  "_id" : "001-003-ABC",
  "first_name" : "John",
  "last_name" : "Doe",
  "email" : "jdoe@doe.com",
  "dob" : "1970-04-23",
  "points" : 55000,
  "interests" : ["sports", "movies"],
  "address" : {
    "street" : "1212 Maple Street",
    "city" : "San Jose",
    "state" : "CA",
    "zip" : "95101"
  },
  "sessions" : [
    {"id":"CoD4","ts":1439824477},
    {"id":"fifa14","ts": 1439565276}
  ]
}
```

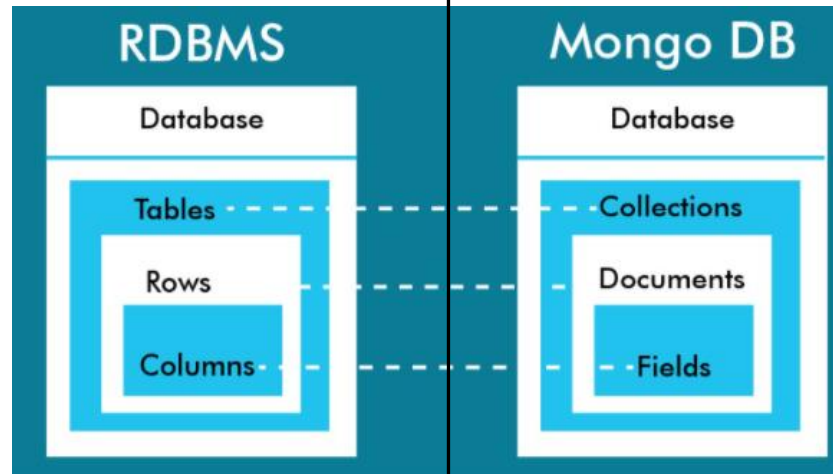
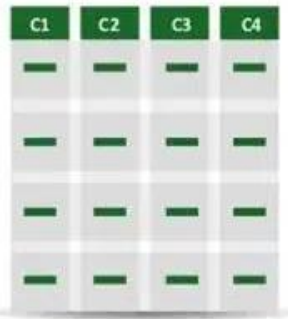
# Flexible Schema



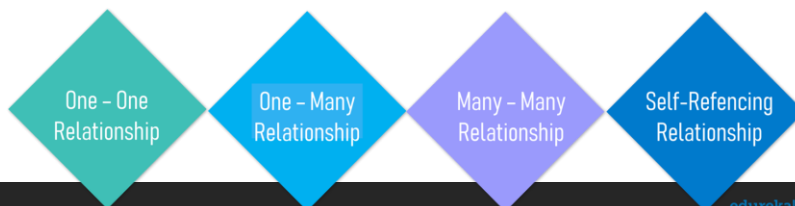
# RDBMS vs. Document - What's the difference

## Relational data model

### A Table

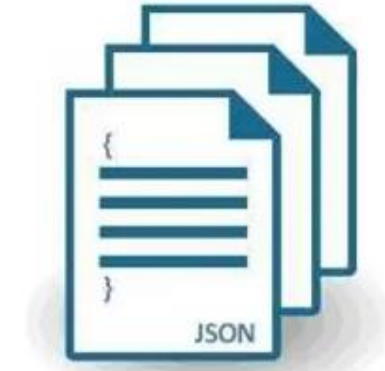


### Relationship : PK, FK, JOIN



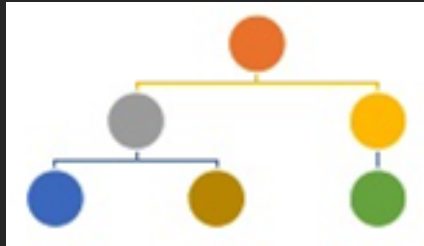
## Document data model

### A Collection



### Relationship: embedded & linking, multi-value field types

# 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

# Flexible Schema in Action







**Best Sellers** View  

Showing 10 items Sort By **Best selling**

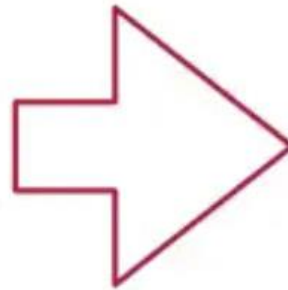
 <b>Vitus Bikes Sommet VR Suspension Bike 2015</b> ★★★★★ 5 Reviews \$2312.99 RRP \$3153.99 <b>SAVE 27%</b> -- SELECT OPTION -- <b>BUY</b>	 <b>Vitus Bikes Escarpe 290 PRO Suspension Bike 2015</b> ★★★★★ 1 Reviews \$3903.49 RRP \$5677.49 <b>SAVE 31%</b> -- SELECT OPTION -- <b>BUY</b>	 <b>Cube Agree GTC Triple Road Bike 2014</b> ★★★★★ 8 Reviews \$1082.99 RRP \$1890.99 <b>SAVE 43%</b> -- SELECT OPTION -- <b>BUY</b>
 <b>Pinarello Marvel T2 Athena Road Bike 2014</b> ★★★★★ \$2891.49 - \$2949.99 RRP \$5204.49 - \$5309.99 <b>SAVE UP TO 44%</b> -- SELECT OPTION -- <b>BUY</b>	 <b>Cube Agree GTC Race Compact Road Bike 2015</b> ★★★★★ 5 Reviews \$2022.49 RRP \$2206.49 <b>SAVE 8%</b> -- SELECT OPTION -- <b>BUY</b>	 <b>Colnago AC-R - Ultegra Road Bike 2015</b> ★★★★★ \$2674.49 RRP \$3942.49 <b>SAVE 32%</b> -- SELECT OPTION -- <b>BUY</b>

**Best Sellers** View  

Showing 1 - 48 of 100 Sort By **Best selling**

 <b>RockShox Reverb Seatpost</b> ★★★★★ 221 Reviews \$269.99 RRP \$460.00 <b>SAVE 41%</b> -- SELECT OPTION -- <b>BUY</b>	 <b>Chain Reaction Cycles Gift Voucher</b> ★★★★★ 88 Reviews \$1.00 - \$200.00 -- SELECT OPTION -- <b>VIEW</b>	 <b>Evoc Bike Travel Bag 280L</b> ★★★★★ 28 Reviews \$450.00 - \$490.00 RRP \$475.00 - \$490.00 <b>SAVE UP TO 5%</b> -- SELECT OPTION -- <b>BUY</b>
 <b>Continental Grand Prix 400S II Road Tyre - 25c PAIR</b> ★★★★★ 27 Reviews \$99.99 RRP \$157.99 <b>SAVE 37%</b> -- SELECT OPTION -- <b>BUY</b>	 <b>RockShox Reverb Stealth Seatpost</b> ★★★★★ 77 Reviews \$314.99 - \$329.99 RRP \$544.00 - \$551.99 <b>SAVE UP TO 42%</b> -- SELECT OPTION -- <b>BUY</b>	 <b>Shimano Ultegra 6700 10 Speed Road Cassette</b> ★★★★★ 160 Reviews \$44.99 RRP \$89.99 <b>SAVE 50%</b> -- SELECT OPTION -- <b>BUY</b>

# Product Catalog - RDBMS



“Entity Value Attribute” Pattern

```
SELECT * FROM (
  SELECT
    ce.sku,
    ea.attribute_id,
    ea.attribute_code,
    CASE ea.backend_type
      WHEN 'varchar' THEN ce_varchar.value
      WHEN 'int' THEN ce_int.value
      WHEN 'text' THEN ce_text.value
      WHEN 'decimal' THEN ce_decimal.value
      WHEN 'datetime' THEN ce_datetime.value
      ELSE ea.backend_type
    END AS value,
    ea.is_required AS required
  FROM catalog_product_entity AS ce
  LEFT JOIN eav_attribute AS ea
    ON ce.entity_type_id = ea.entity_type_id
  LEFT JOIN catalog_product_entity_varchar AS ce_varchar
    ON ce.entity_id = ce_varchar.entity_id
    AND ea.attribute_id = ce_varchar.attribute_id
    AND ea.backend_type = 'varchar'
  LEFT JOIN catalog_product_entity_text AS ce_text
    ON ce.entity_id = ce_text.entity_id
    AND ea.attribute_id = ce_text.attribute_id
    AND ea.backend_type = 'text'
  LEFT JOIN catalog_product_entity_decimal AS ce_decimal
    ON ce.entity_id = ce_decimal.entity_id
    AND ea.attribute_id = ce_decimal.attribute_id
    AND ea.backend_type = 'decimal'
  LEFT JOIN catalog_product_entity_datetime AS ce_datetime
    ON ce.entity_id = ce_datetime.entity_id
    AND ea.attribute_id = ce_datetime.attribute_id
    AND ea.backend_type = 'datetime'
  WHERE ce.sku = 'rp-prod132546'
) AS tab
WHERE tab.value != '';
```

**To get a single product**

# Product Catalog – NoSQL/Document

```
{
  "_id" : "rp-prod132546",
  "name" : "Marvel T2 Athena",
  "brand" : "Pinarello",
  "category" : "bike",
  "type" : "Road Bike",
  "price" : 2949.99,

  "size" : "55cm",
  "wheel_size" : "700c",
  "frameset" : {
    "frame" : "Carbon Toryaca",
    "fork" : "Onda 2V C"
  },
  "groupset" : {
    "chainset" : "Camp. Athena 50/34",
    "brake" : "Camp."
  },
  "wheelset" : {
    "wheels" : "Camp. Zonda",
    "tyres" : "Vittoria Pro"
  }
}
```



```
products
  .findById("rp-prod132546")
```

Store the product “as a business object”

**To get a single product**

# Easy Variation with Document

```
{
  "_id" : "rp-prod132546",
  "name" : "Marvel T2 Athena",
  "brand" : "Pinarello",
  "category" : "bike",
  "type" : "Road Bike",
  "price" : 2949.99,

  "size" : "55cm",
  "wheel_size" : "700c",
  "frameset" : {
    "frame" : "Carbon Toryaca",
    "fork" : "Onda 2V C"
  },
  "groupset" : {
    "chainset" : "Camp. Athena 50/34",
    "brake" : "Camp."
  },
  "wheelset" : {
    "wheels" : "Camp. Zonda",
    "tyres" : "Vittoria Pro"
  }
}
```

```
{
  "_id" : "rp-prod106702",
  "name" : " Ultegra SPD-SL 6800",
  "brand" : "Shimano",
  "category" : "pedals",
  "type" : "Components",
  "price" : 112.99,

  "features" : [
    "Low profile design increases ...",
    "Supplied with floating SH11 cleats",
    "Weight: 260g (pair)"
  ]
}
```

```
{
  "_id" : "rp-prod113104",
  "name" : "Bianchi Pride Jersey SS15",
  "brand" : "Nalini",
  "category" : "Jersey",
  "type" : "Clothing",
  "price" : 76.99,

  "features" : [
    "100% Polyester",
    "3/4 hidden zip",
    "3 rear pocket"
  ],
  "color" : "black"
}
```

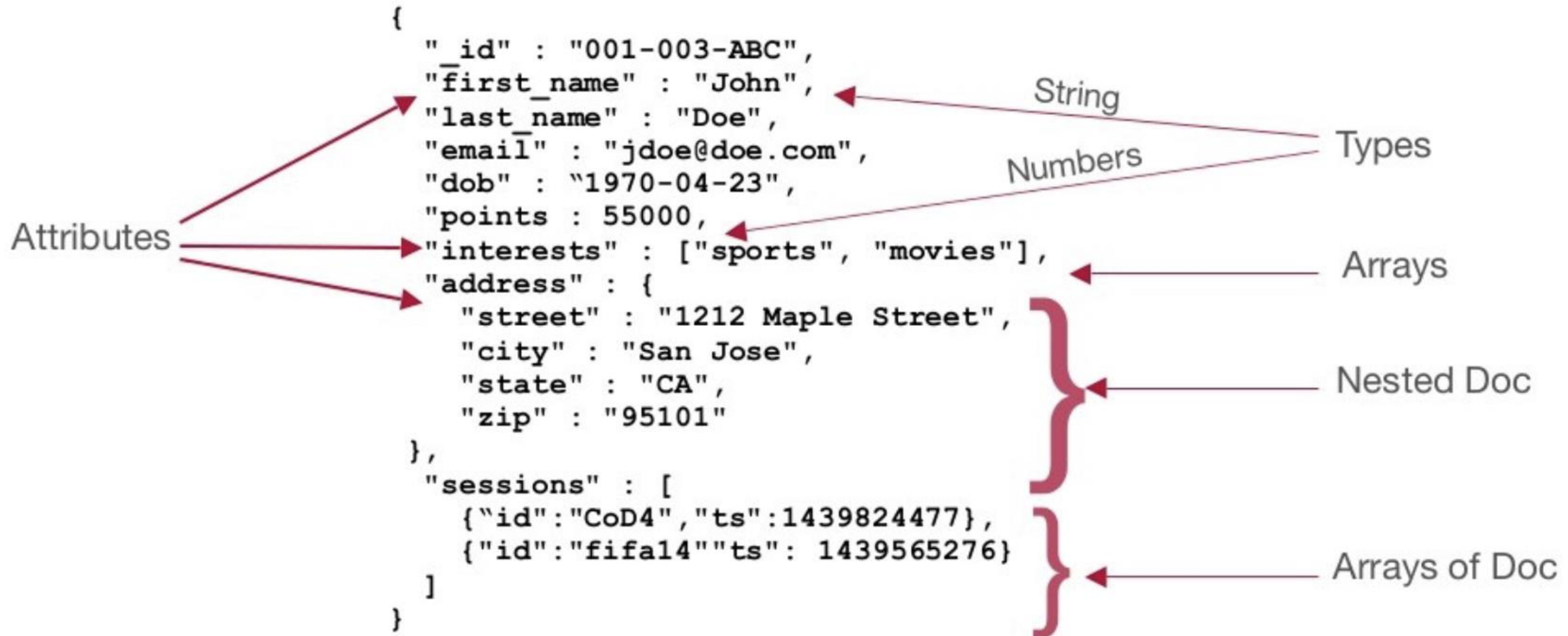


# Flexible Schema

- Support Data Types
- Complex Data Structure
- Developer Friendly
- Flexible

```
{
  "_id" : "001-003-ABC",
  "first_name" : "John",
  "last_name" : "Doe",
  "email" : "jdoe@doe.com",
  "dob" : "1970-04-23",
  "points" : 55000,
  "interests" : ["sports", "movies"],
  "address" : {
    "street" : "1212 Maple Street",
    "city" : "San Jose",
    "state" : "CA",
    "zip" : "95101"
  },
  "sessions" : [
    {"id":"CoD4","ts":1439824477},
    {"id":"fifa14","ts": 1439565276}
  ]
}
```

# Flexible Schema



## Add feature easily

Requirement: "users can vote and comment product"

```
{
  "_id" : "rp-prod113104",
  "name" : "Bianchi Pride Jersey SS15",
  "brand" : "Nalini",
  "category" : "Jersey",
  "type" : "Clothing",
  "price" : 76.99,
  "features" : [
    "100% Polyester",
    "3/4 hidden zip",
    "3 rear pocket"
  ],
  "color" : "black",
  "comments" : [{...}, {...}],
  "ratings" : [{..., "value" : 5}, {..., value : 3}],
  "rating" : 4
}
```

**Done: just store the data!**  
***Means less time to market***

# 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

---





# 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

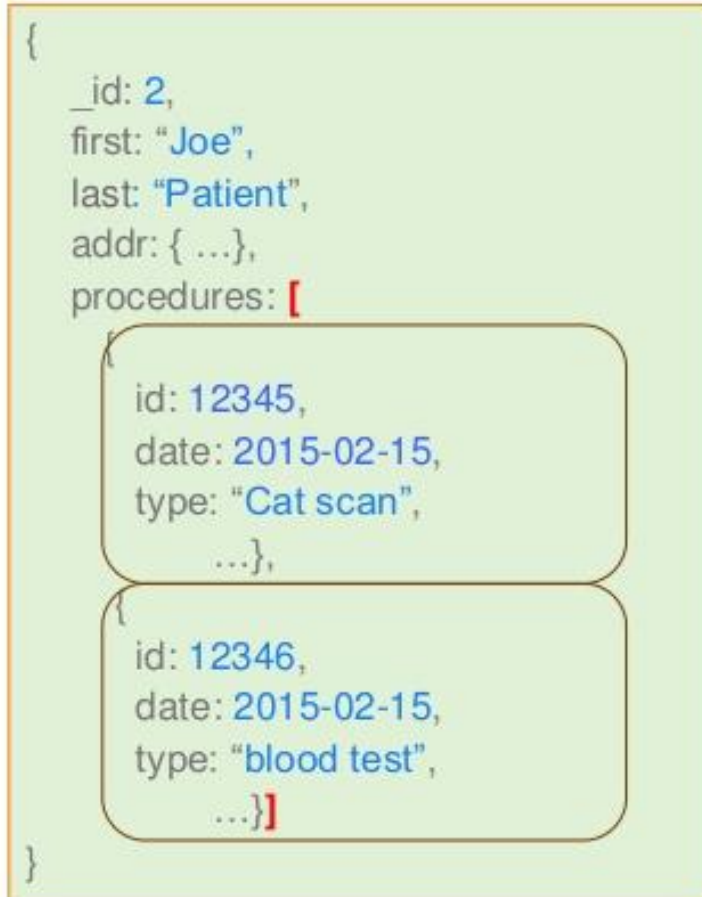
---

# 1-M

Modeled in 2 possible ways

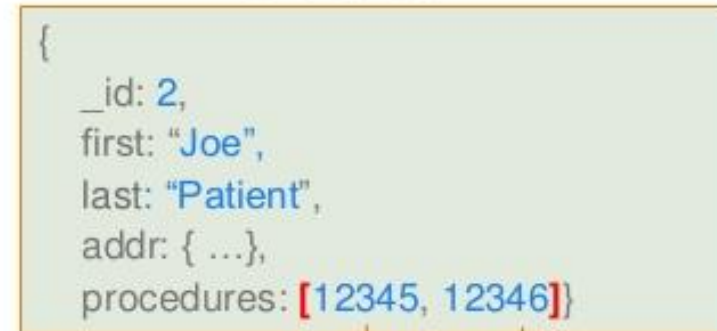
## Embed

Patients

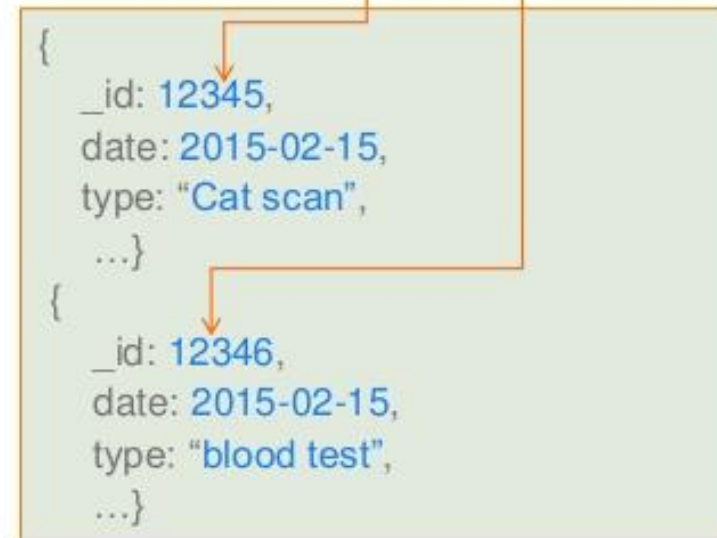


## Reference

Patients



Procedures





## 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

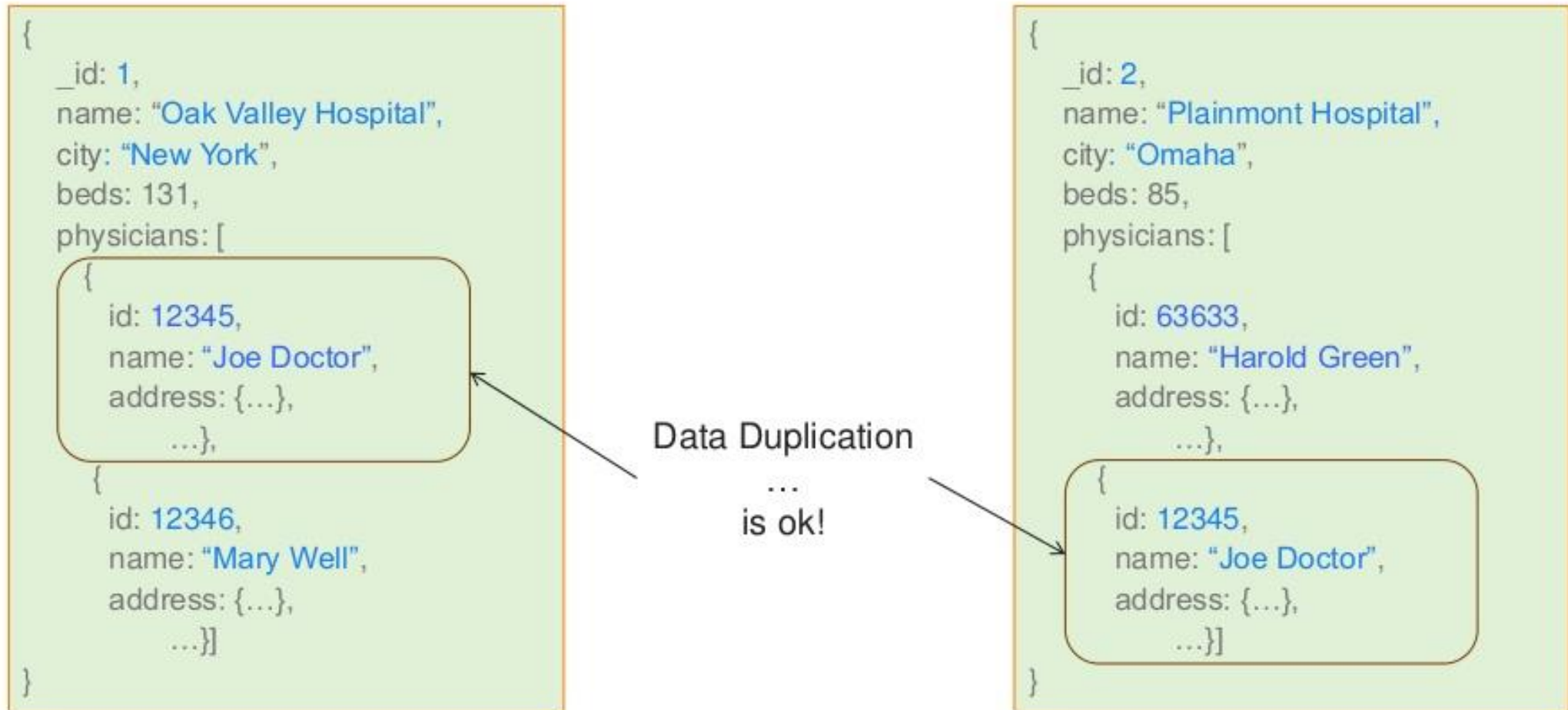
```
{
  _id: 2,
  first: "Joe",
  last: "Patient",
  addr: { ...},
  procedures: [
    {
      id: 12345,
      date: 2015-02-15,
      type: "Cat scan",
      ...},
    {
      id: 12346,
      date: 2015-02-15,
      type: "blood test",
      ...}]
}
```

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>]