Linked Data

From Document to Graph ?
From the Mango to the Avocado !

Or a feedback on the Mango

By Michaël DOHR
27/02/2019

Plan

1

Introduction

1. Presentation
2. Why use a document database?

2

Data design in Document DB

1. Concepts
2. Relationships

3

Data design in Pure Graph DB

4

Hybrid database ?

Introduction

Why use a document database?

Presentation

Data is increasing in volume …

- New digital process
- More online transaction
- New Social networks
- More devices

... and is getting more connected.

Customers, products, devices interact and related to each other

Using Data Relationships unlocks value

- Real-time recommendations
- Fraud detections
- Master data management
- Network and IT operations
- Identity and access management

Why document database ?

Flexible data modelling

- Schema less
- Embedded related data
- Easy to migrate, to change
- Easy to restore

Query engine

- Aggregation and data computation
- Cost to cost result (as Json)

HA (High Availability)

- Replication by design
- Data sharding

```
{
        "_id": "movie-1",
        "title": "The Avengers",
        "translatedTitle": {
                "en": "The Avengers",
                "fr": "The Avengers"
        },
        "rating": 8.1,
        "releaseYear": "2012",
        "suspended": false,
        "genres": [ "Action", "Aventure", "Sci-Fi" ],
        "country": "USA",
        "roles": [
                {
                        "name": "Tony Stark / Iron Man",
                        "actor": {
                                "id": "651778c4-5105-47a7",
                                "name": "Robert Downey Jr."
                        }
                },
                {
                        "name": "Natasha Romanoff / Black Widow",
                        "actor": {
                                "id": "a80906da-3ede-4c60",
                                "name": "Scarlett Johansson"
                        }
                }
        ]
}
```

Data design in Doc DB

Concepts, Relationships

Concepts

- **Schema less != Anarchy**
  - Schema should be controlled by the application
- **Document should be self sufficient**
- **Final document size has an impact to performance.**
- **Specifics to MongoDB**
  - Relationships is more complicated than SQL Database or ArangoDB
  - Need to unnormalize data

```
{
        "_id": "movie-1",
        "title": "The Avengers",
        "translatedTitle": {
                "en": "The Avengers",
                "fr": "The Avengers"
        },
        "rating": 8.1,
        "releaseYear": "2012",
        "suspended": false,
        "genres": [ "Action", "Aventure", "Sci-Fi" ],
        "country": "USA",
        "roles": [
                {
                        "name": "Tony Stark / Iron Man",
                        "actor": {
                                "id": "651778c4-5105-47a7",
                                "name": "Robert Downey Jr."
                        }
                },
                {
                        "name": "Natasha Romanoff / Black Widow",
                        "actor": {
                                "id": "a80906da-3ede-4c60",
                                "name": "Scarlett Johansson"
                        }
                }
        ]
}
```

Relationships

- **1..1**
  - Directly embedded in the document
- **1..n**
  - Embedded
  - By Reference
- **n..n**
  - By Reference in one or both collections

```
{
        "title": "Comparison between Document DB and Graph DB",
        "localizedTitle": {
                "en": "Comparison between Document DB and Graph DB"
        },
        "authorId": "DOH",
        "body": "Lorem ipsum dolor sit amet, ...",
        "publishedAt": "2019-02-27T17:30:00.000Z",
        "comments": [
                {
                        "body": "😊",
                        "approved": true,
                        "at": "2019-02-27T18:30:00.000Z",
                        "author": {
                                "id": "WAB",
                                "name": "Bruce Wayne"
                        }
                }
        ],
        "likedBy": ["WAB", "CNR"]
}
```

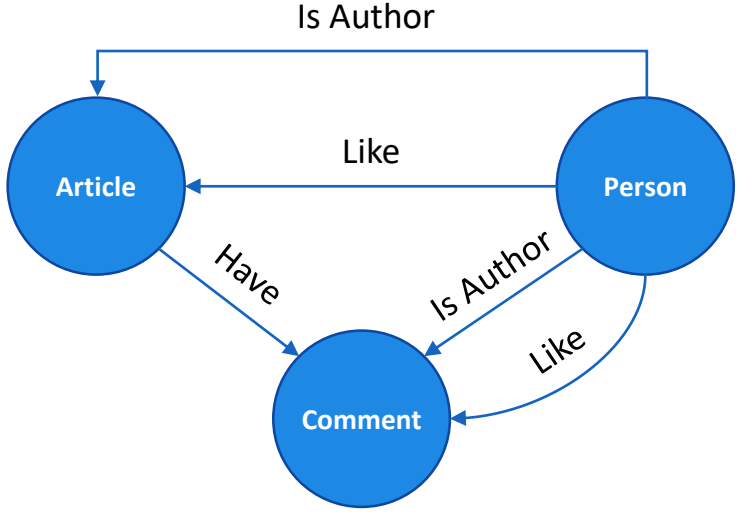
Conclusion

- **In most of the cases the conception is API oriented**
  - Primary data is in the main document
  - Lazy list of values in dedicated collections
- **Entity relations works perfectly with class diagram (UML)**
  - External links are implemented by **Aggregation**
  - Embedded links are implemented by **Composition**
- **Collections can be associated to SQL Table (for data segregation)**
  - But without Table constraints
- **Queries can be really complexes when you have a lot of relations.**
  - Like SQL Database
  - But you can easily delegate this complexity to another query system, like GraphQL

Data design in Graph DB

Concepts

- **Reference database : Neo4j**
  - In Graph, we discuss about Vertex and Edge
  - Vertex materialize Entities and Edge materialize Relationships
  - Both of them have properties
  - Properties are primitive or array of primitive.



Hybrid database ?

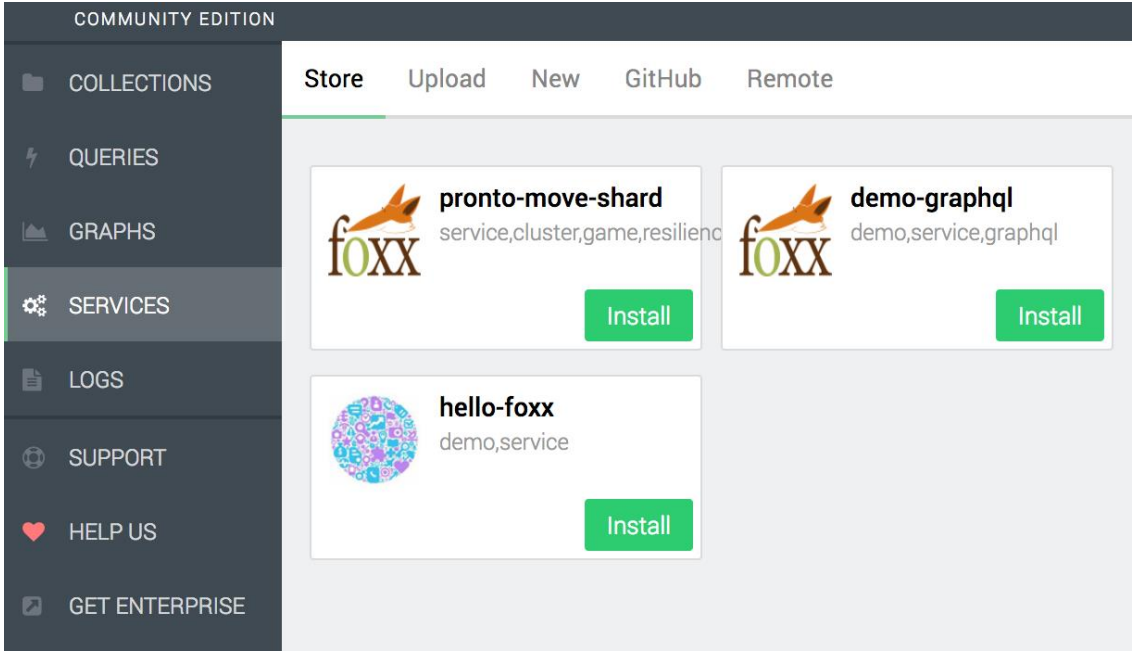
Comparison

- **Document Database (MongoDB)**
  - **Pros**
    - Embedded sub-documents
    - Cost to Cost API
    - Aggregation and data transformation
    - Sharding and replication
    - Segregation by collections
  - **Cons**
    - Not transactional (< 4.0)
    - Relation is complicated to implement

- **Graph Database (Neo4j)**
  - **Pros**
    - Transactional
    - Powerful relation model
    - Graph algorithm
    - Query language (Cypher)
  - **Cons**
    - Node contains only primitive fields
    - One universe (database) for all kind of data.

ArangoDB

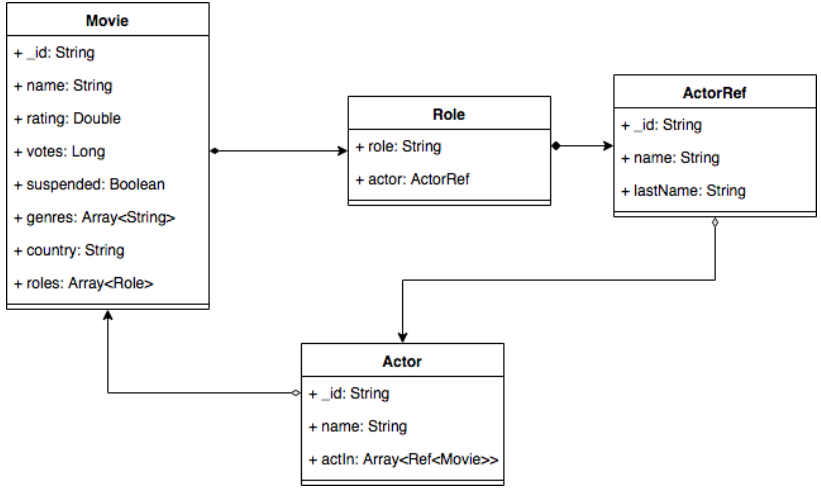
- **Multi model database**
  - Document
  - Graph
- **AQL language**
- **Extensible (via Foxx)**



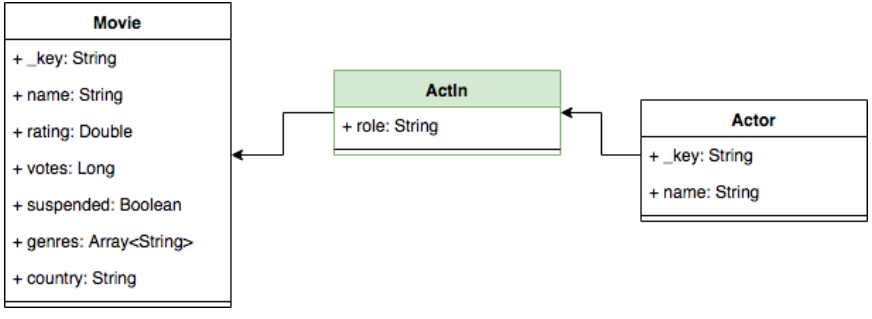
Examples

Examples

- **MongoDB Design**



- **ArangoDB Design**



Examples

- **Find “Batman Forever”.**
- **Find all “Batman” movies.**
- **Find “Kevin Spacey” best note.**
- **Count movies by actor.**
- **Find movies category for the actor who played in the most movies.**
- **Find all actors of “Batman Forever”.**
- **Find all co-actors of “Kevin Spacey”.**
- **Find all actors who did not play with “Kevin Bacon”.**
- **Calculate rating average.**
- **Top 10 of best movies.**
- **Top 10 of worst movies.**

Thank You

Questions?

References

- **Market share :**
  - Graph : https://db-engines.com/en/ranking/graph+dbms
  - Document : https://db-engines.com/en/ranking/document+store
- **Benchmark**
  - https://dzone.com/articles/nosql-performance-benchmark-2018-mongodb-postgresq
- **Drivers**
  - Golang : https://github.com/arangodb/go-driver
  - Java : https://github.com/arangodb/arangodb-java-driver
  - Spring-Data : https://github.com/arangodb/spring-data
- **Neo4j presentation**
  - https://github.com/dohr-michael/neo4j_sid