Systems and Methods for Big and Unstructured Data Project

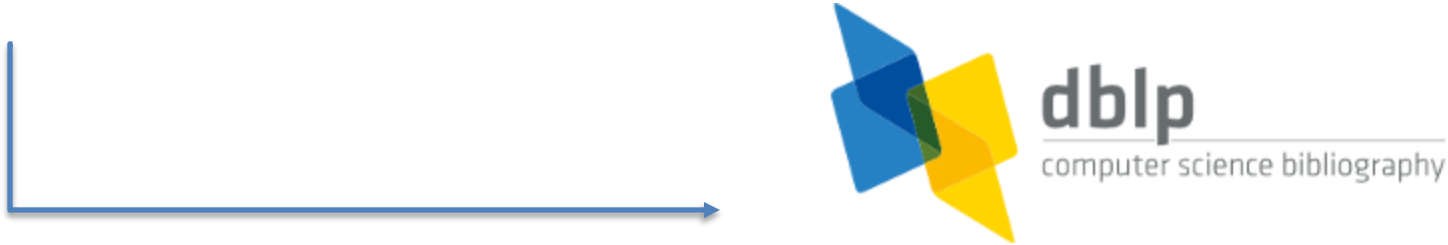
Lorenzo Biasiolo – 10629367
Grecya D'Angiò - 10651939
Elia Maggioni – 10610008
Enrico Maria Marinelli – 10898730
Carlos Santillán - 10659783

Group 7

Academic Year 2022-2023

1. Problem presentation and assumptions:

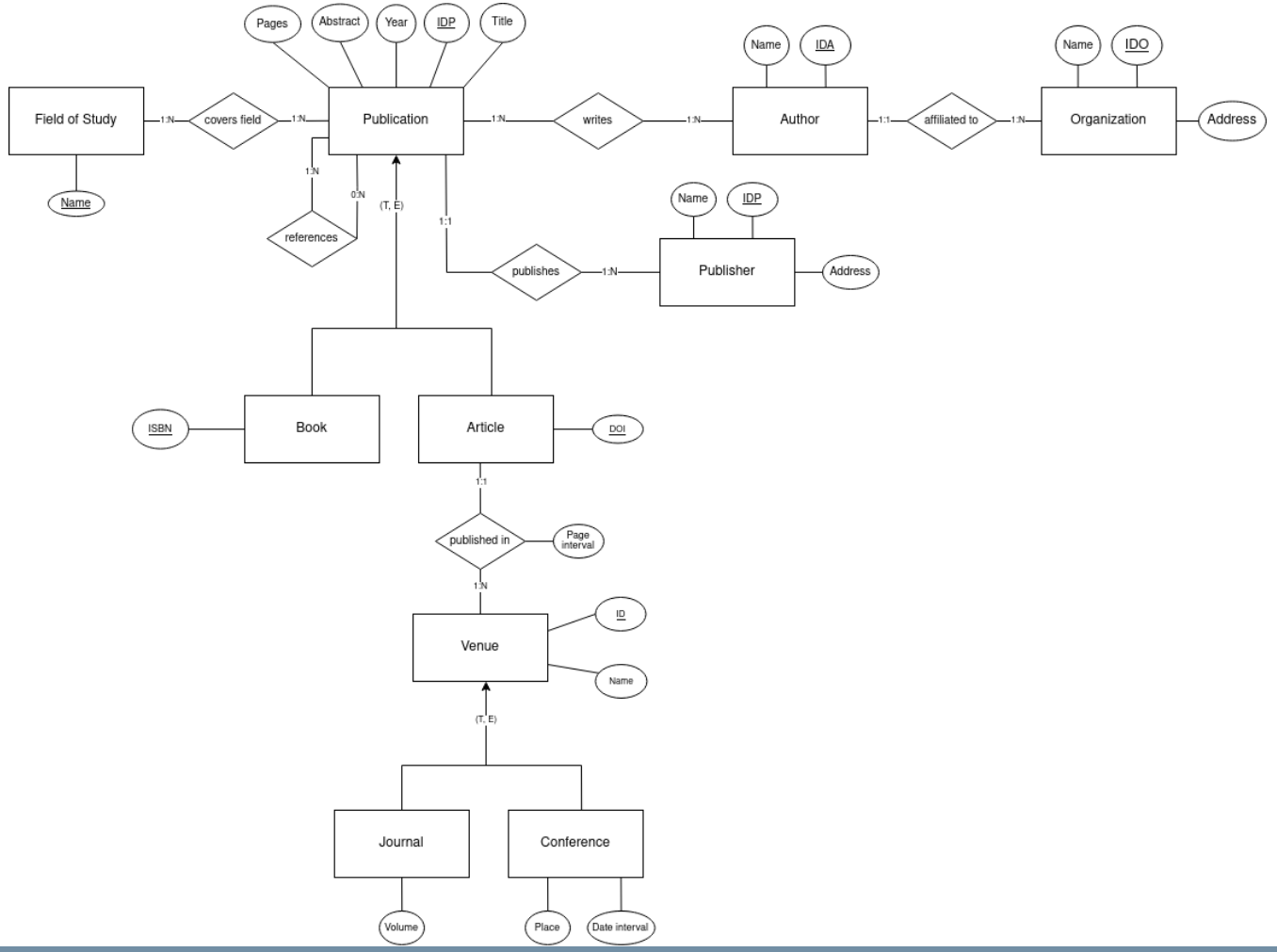
- Bibliography database
- DBLP website



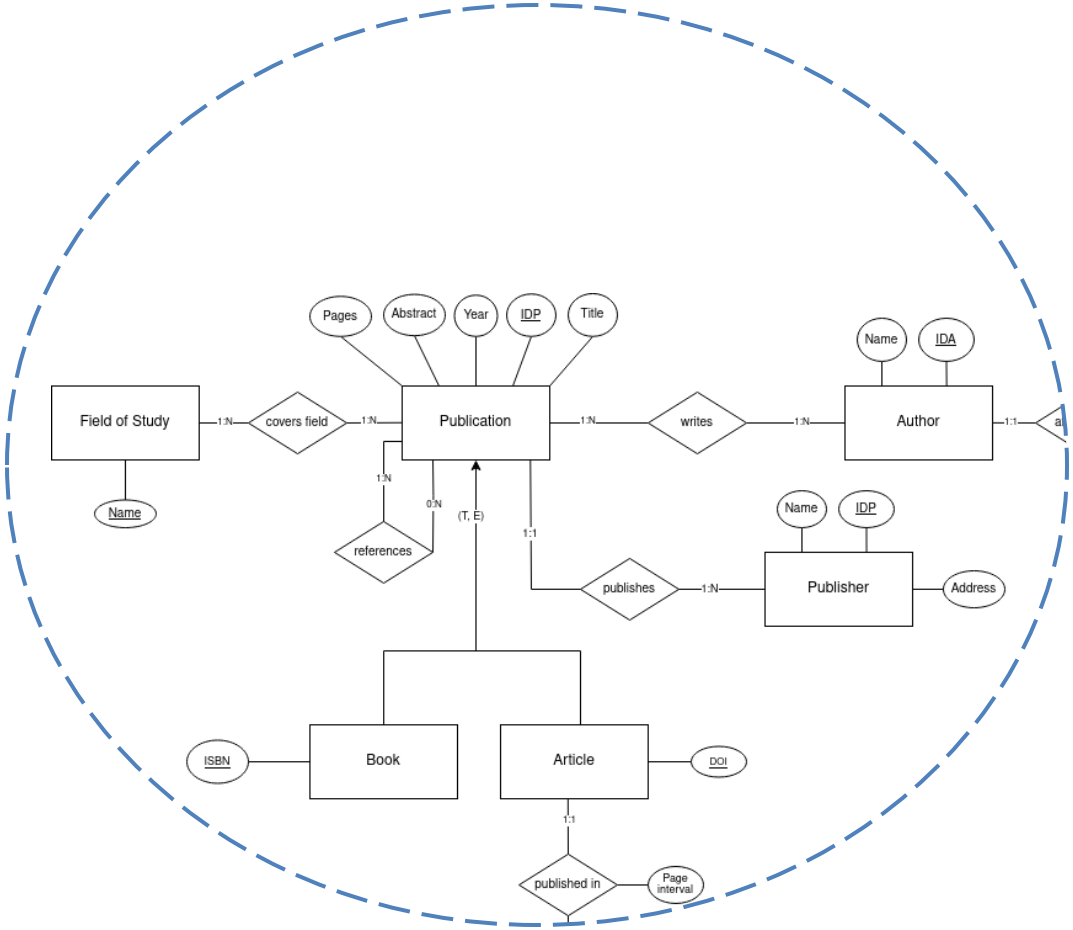
1. Problem presentation and assumptions:

- An author can be affiliated to one and only one organization.
- There is no distinction between the authors of the same publication, no ranking nor order. We assume they all have contributed equally to the publication.
- A publication X must cite at least one other publication, but it is admissible that no one has cited (referenced) X.
- There are no joint publishers (i.e., publishers that share publication rights for the publications). Each publication must have one and only one publisher (even if it is presented at a conference).
- Publications stored in the database can either be books or articles. No other types exist. It could eventually be expanded to include works like Ph.D. theses and independent publications.
- Each article must be presented in one venue regardless of the type of venue.
- A publication cannot reference itself.
- Whenever we see missing values in the dataset, we set them to a default value.

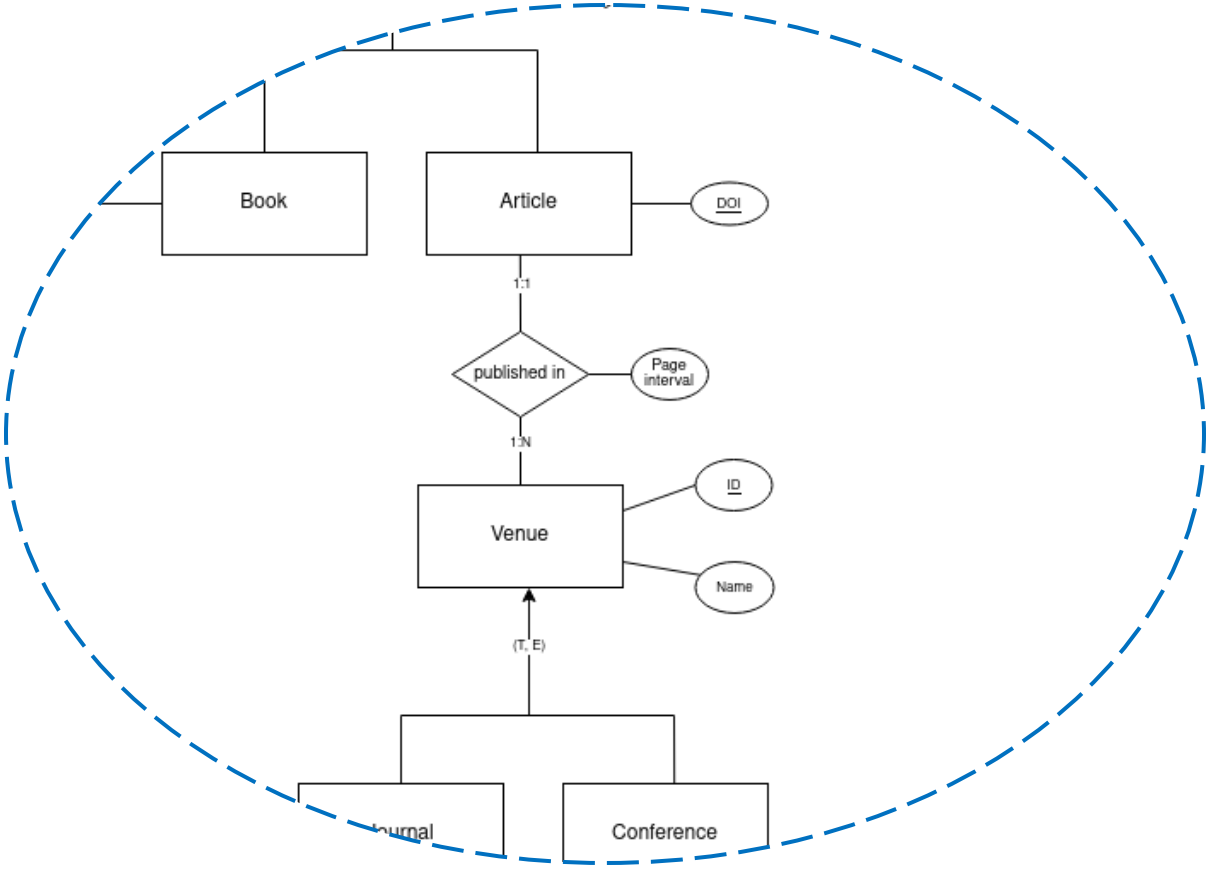
2. ER Diagram


Pages
Abstract
Year
IDP
Title
Field of Study
1:N
covers field
1:N
Publication
1:N
writes
1:N
Author
Name
IDA
1:1
affiliated to
1:N
Organization
Name
IDO
Address
Name
1:N
0:N
references
(T, E)
1:1
publishes
1:N
Publisher
Name
IDP
Address
ISBN
Book
Article
DOI
1:1
published in
Page interval
1:N
Venue
ID
Name
(T, E)
Journal
Conference
Volume
Place
Date interval

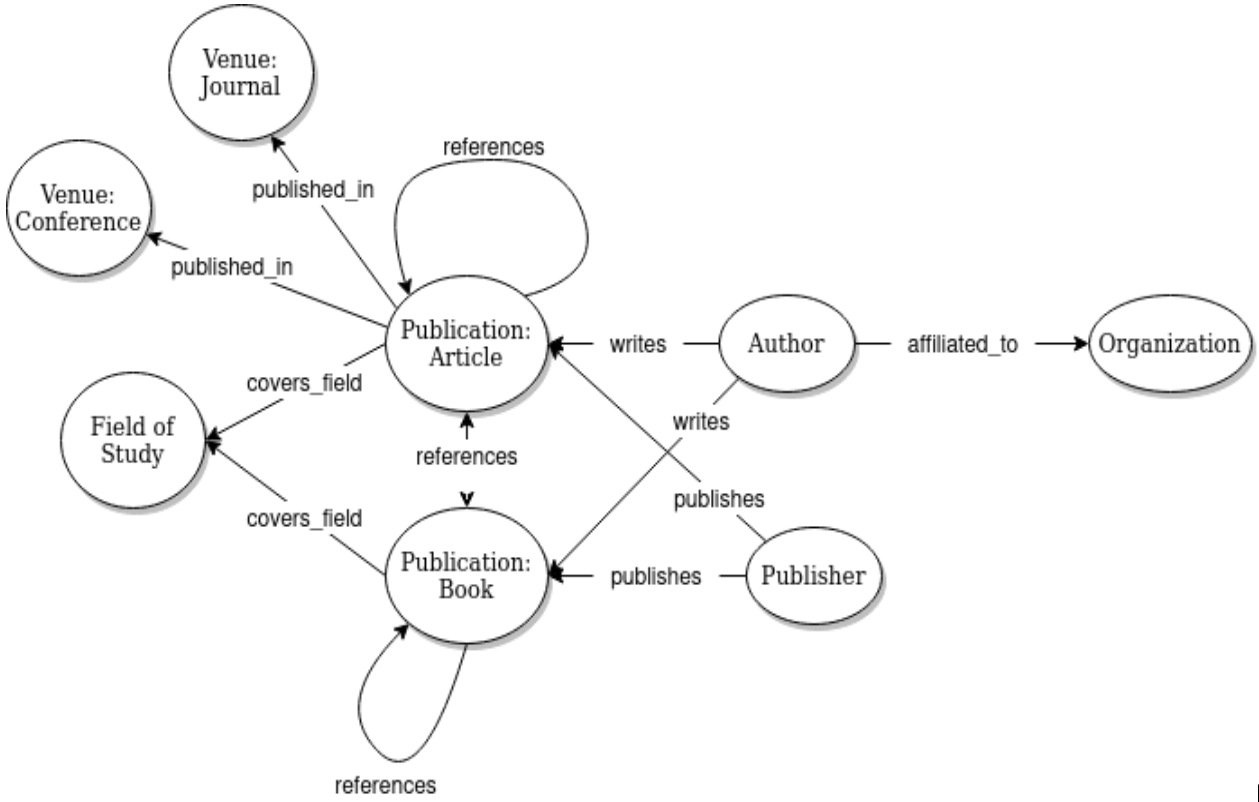
2. ER Diagram


Pages
Abstract
Year
IDP
Title
Name
IDA
Field of Study
covers field
Publication
writes
Author
Name
1:N
0:N
(T, E)
1:1
references
Name
IDP
publishes
Publisher
Address
ISBN
Book
Article
DOI
published in
Page interval

2. ER Diagram


Book
Article
DOI
1:1
published in
Page interval
1:N
Venue
ID
Name
(T, E)
Conference

3. Neo4J



Useful to research the relationships of the different parts of the schema

3.1 Neo4J Dataset structure

The dataset had a .json format and had the following structure for each item.

```
[
    {
        "id": 2456,
        "authors": [{"name": "Mario Rossi", "id": 4,
                     "org": "Politecnico di Milano"}],
        "title": "Non-cooperative games",
        "abstract": "This is an awesome report, here is why"
        "year": 2005,
        "page_start": 245,
        "page_end": 255,
        "doc_type": "article",
        "publisher": "Springer",
        "fields": [{"name": "Computer science"},
                   {"name": "Artificial intelligence"}],
        "venue": {"raw": "Game theory journal", "id": 235, "type": "J"}
    }
    {
        "id": 2453
        ...
        ...
    }
    ...
]
```

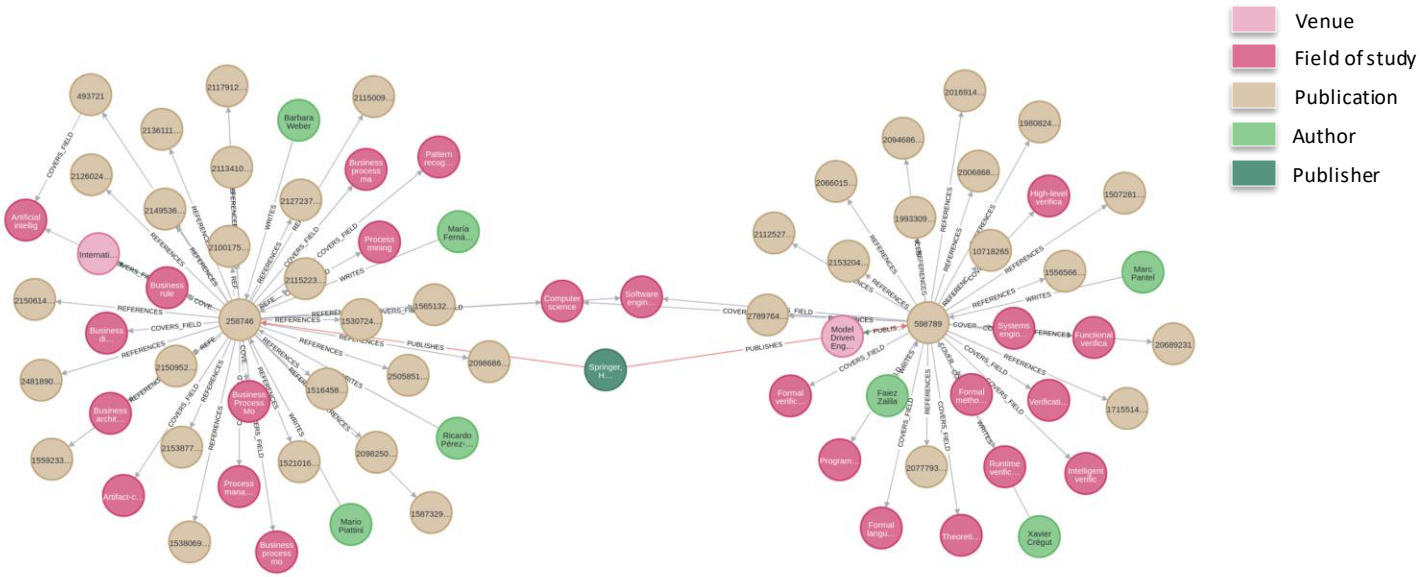
3.2 Query

Find the articles (with at least 12 pages) that cover the most fields by the publisher who has published the most articles (assumed to be unique).

```
MATCH (publisher:Publisher)-[pub:PUBLISHES]->(:Article)
WITH COUNT(pub) AS publications, publisher
ORDER BY publications DESC LIMIT 1
WITH publisher
MATCH (publisher)-[:PUBLISHES]->(a:Article)-[c:COVERS_FIELD]->(:FieldOfStudy)
WHERE a.pages >= 12
WITH COUNT(c) AS fields, a, publisher
WITH MAX(fields) AS max_fields, publisher
MATCH (publisher)-[:PUBLISHES]->(a:Article)-[c:COVERS_FIELD]->(:FieldOfStudy)
WITH COUNT(c) AS fields, a, max_fields
WHERE fields = max_fields AND a.pages >= 12
RETURN a
```

3.2 Results

Find the articles (with at least 12 pages) that cover the most fields by the publisher who has published the most articles (assumed to be unique)

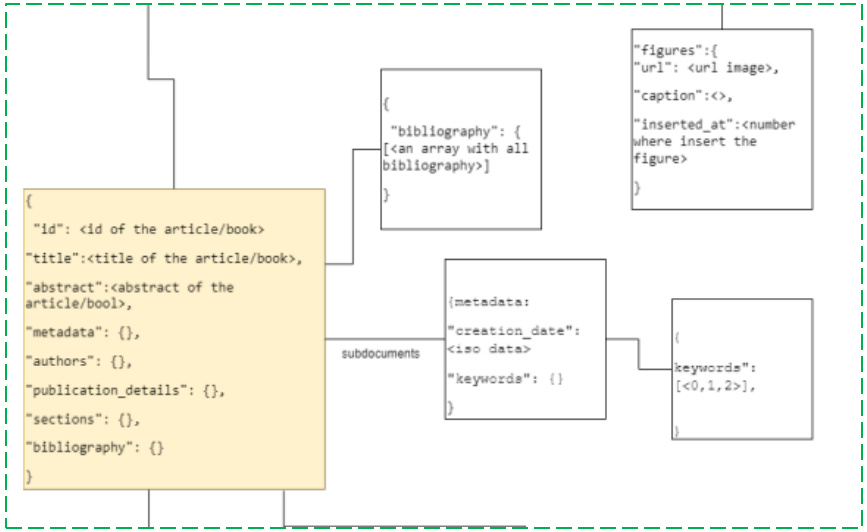


4. MongoDB

Useful to store elements together, flexible structure

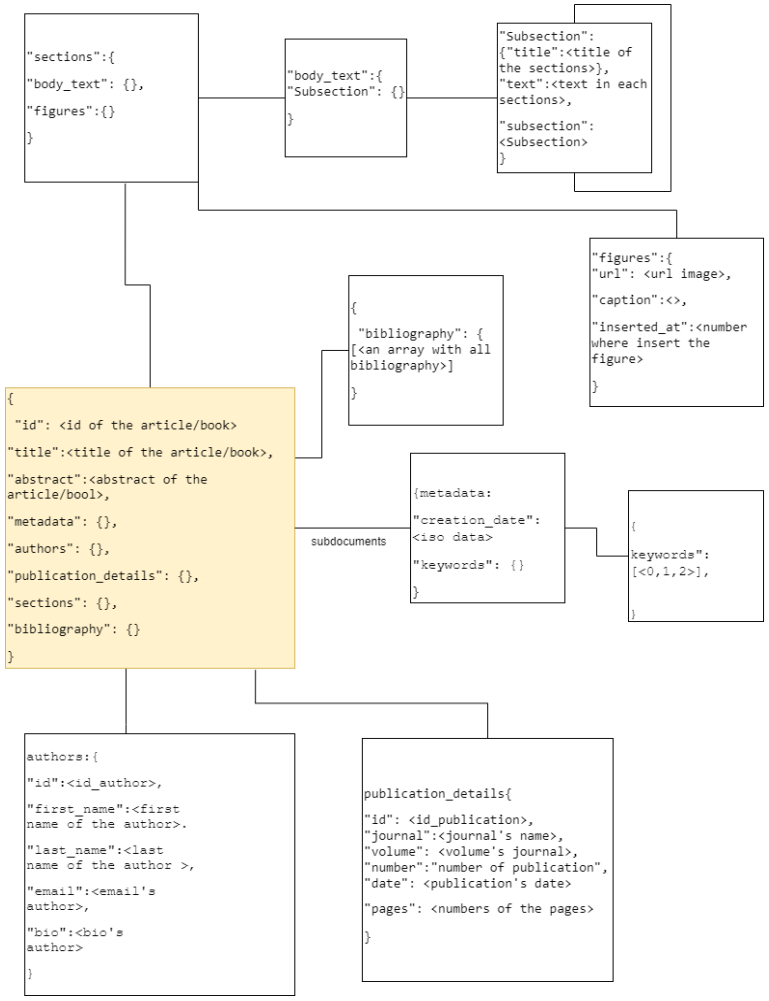
Documental DB

mongoDB®



How do we create the collection?

4.1 Document diagram



4.1 MongoDB Data Structure


```
{
        "id": 0,

        "title": "Amphibolurus barbatus",

        "abstract": "Duis bibendum, felis sed interdum venenatis,
                     turpis enim blandit mi, in porttitor pede justo eu massa
                     Donec dapibus. Duis at velit eu est congue elementum.",
        "metadata": {
                     "creation_date": {"$date":"1982-06-29T07:31:02.000Z"},
                     "keywords":["Visionoriented","disintermediate",
                                 "system","engine"]},
        "authors": [{
                     "id":206,
                     "first_name":"Kristina",
                     "last_name":"Gapp",
                     "email":"kgapp5p@noaa.gov",
                     "bio":"REMEDYREPACKINC.",
                     "affiliation":"Katz"}],


         "publication_details":
                  [{
                    "id":24,
                    "journal":"Ecole Normale
                               Superieure de Fontenay-Saint Cloud",
                    "volume":"Fabaceae",
                    "number":"74-963-4668",
                    "date": {"$date":"1995-11 01T22:48:12.980Z"},
                    "pages":49}],
                    "sections":
                       [{
```

```
"body_text":{
   "title":"Synchronised cohesive algorithm",
   "text":[
      "Pork belly kevin andouille prosciutto.",
   ],
   "sub_section":[{
         "title":"Optimized leading
                  edge customer loyalty",
         "text":[
            "Salami tongue ham hock",
         ],
         "sub_section":[{
             "title":"Advanced fault-tolerant
                      matrices",
             "text":[
                "Prosciutto pig pork belly",
             ]}
         ]
      },
   ]
},
```

4.1 MongoDB Data Structure


```
            "figures":[
               {
                  "url":"http://dummyimage.com/237x100.png",
                  "caption":null,
                  "inserted_at":6405
               },
               {
                  "url":"http://dummyimage.com/115x100.png",
                  "caption":"Innovative human-resource
                             knowledge user",
                  "inserted_at":7906
               }
            ]
         }],

"bibliography":[41,77,148,191,217,299,335,364,441,452,512]}
```

4.2 Query

Find authors id who have more than 140 publications

```
db.articles.aggregate([
   {
      "$unwind":{
         "path":"$authors"
      }
   },
   {
      "$group":{
         "_id":"$authors.id",
         "first_name":{
            "$addToSet":"$authors.first_name"
         },
         "last_name":{
            "$addToSet":"$authors.last_name"
         },
         "publications":{
            "$push":"$publication_details.id"
         }
      }
   },
   {
      "$project":{
         "publications":{
            "$reduce":{
               "input":"$publications",
               "initialValue":[

               ],
               "in":{
                  "$concatArrays":[
                     "$$value",
                     "$$this"
                  ]
               }
            }
         },
  "first_name":"$first_name",
  "last_name":"$last_name",
      }
   },
{$match: {$expr:{$gt:[{$size:"$publications"}, 140]}}}
])
```

4.2 Results

Find authors id who have more than 140 publications


```
{ _id: 219,
  publications:
   [ 12,
     91,
     137,
     166,
     267,
     292,
     382,
     416,
     445,
     538,
     567,
     631,
```

5. Spark Data Structure

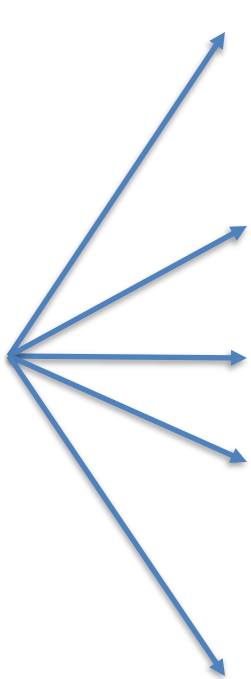
```
schemapub = StructType([ \
    StructField('id', StringType(), False), \
    StructField('title', StringType(), False), \
    StructField('page_start', IntegerType(), True), \
    StructField('page_end', IntegerType(), True), \
    StructField('year', IntegerType(), True), \
    StructField('citations', IntegerType(), True), \
    StructField('venue', StringType(), True), \
    StructField('keywords', StringType(), True) \
])

schemaref = StructType([ \
    StructField('references', StringType(), False), \
    StructField('referenced', StringType(), False) \
])

schemavenue = StructType([ \
    StructField('name', StringType(), False), \
    StructField('type', StringType(), True) \
])

schemawrites = StructType([ \
    StructField('author', StringType(), False), \
    StructField('publication', StringType(), False) \
])

schemaauthor = StructType([ \
    StructField('id', StringType(), False), \
    StructField('name', StringType(), False), \
    StructField('org', StringType(), True) \
])
```



publication.csv

reference.csv

venue.csv

writes.csv

author.csv

- Fast
- Easy
- General
- Compute huge quantity of data
- Columnar approach

5.1 Query

Find publications that are referenced more than the average that contain the keywords 'Data Mining' and 'Computer Science', ordered by the number of publications in ascending order.

```
average = ref.groupBy('referenced').count() \
                .sort('count',ascending=False) \
                .groupBy() \
                .avg('count') \
                .collect()[0][0]

pub.join(ref,ref.referenced == pub.id, 'left') \
        .select('title', 'id', 'referenced', 'references', 'keywords') \
        .groupBy('title', 'keywords') \
        .agg(count('references').alias('number_of_references')) \
        .filter((col('number_of_references') > average) & \
                array_contains(pub.keywords, 'Computer science') & \
                array_contains(pub.keywords, 'Data mining')) \
        .sort('number_of_references',ascending = True) \
        .show()
```

5.1 Query

Find publications that are referenced more than the average that contain the keywords 'Data Mining' and 'Computer Science', ordered by the number of publications in ascending order.

title	keywords	number_of_references
Modularizing Onto...	[Distributed know...	13
Modeling for Opti...	[Data mining, Com...	13
TOWARDS AN EXTEND...	[Information syst...	15
Rough Sets-Based ...	[Data mining, Equ...	16
Finite model theo...	[Data mining, Fin...	17
DART: an efficien...	[Query optimizati...	18
A Minimum Descrip...	[Bottleneck, Grap...	18
A search-engine c...	[Data mining, Fea...	19
Association Rules...	[Data mining, Com...	19
Two Evolution Ind...	[Linear equation,...	19
Bridging the gaps...	[Semi-structured ...	20
A Dynamical Syste...	[Data mining, Com...	20
An algorithm for ...	[Anomaly detectio...	21
Computing the Spl...	[Decision tree, D...	21
The GOLD Model CA...	[Data warehouse, ...	21
A cluster-based a...	[Design elements ...	22
Multi-view Metric...	[Data point, Auto...	22
Neural Net Agent ...	[Data mining, Wor...	26
Spectral clusteri...	[Hierarchical clu...	27
Fully utilize fee...	[Data mining, Sem...	27

only showing top 20 rows