

Soient quelques instances de métadonnées relatives à des publications scientifiques parues dans deux types de support : revue et conférence, respectivement. Source : <https://dblp.uni-trier.de>.

**Articles de revues** (format BibTeX)

<pre>@article{TruicaADA21, author = {Ciprian-Octavian Truica and Elena Simona Apostol and Jérôme Darmont and Ira Assent}, title = {TextBenDS: a Generic Textual Data Benchmark for Distributed Systems}, journal = {Information System Frontiers}, year = {2021} }</pre>	<pre>@article{SawadogoD21a, author = {Pegdwendé Nicolas Sawadogo and Jérôme Darmont}, title = {On data lake architectures and metadata management}, journal = {Journal of Intelligent Information Systems}, year = {2021} }</pre>
--	---

**Articles de conférences** (format BibTeX)

<pre>@inproceedings{YangDRT22, author = {Yuzhao Yang and Jérôme Darmont and Franck Ravat and Olivier Teste}, title = {Dimensional Data KNN-Based Imputation}, booktitle = {26th ADBIS European Conference}, year = {2022} }</pre>	<pre>@inproceedings{SawadogoD21b, author = {Pegdwendé Nicolas Sawadogo and Jérôme Darmont}, title = {Benchmarking Data Lakes Featuring Structured and Unstructured Data with DLBench}, booktitle = {23rd DaWaK International Conference}, year = {2021} }</pre>
<pre>@inproceedings{AzriFHDN21, author = {Abderrazek Azri and Cécile Favre and Nouria Harbi and Jérôme Darmont}, title = {Calling to CNN-LSTM for Rumor Detection: A Deep Multi-channel Model for Message Veracity Classification in Microblogs}, booktitle = {ECML-PKDD European Conference}, year = {2021} }</pre>	<pre>@inproceedings{YangADRT21, author = {Yuzhao Yang and Fatma Abdelhédi and Jérôme Darmont and Franck Ravat and Olivier Teste}, title = {Internal Data Imputation in Data Warehouse Dimensions}, booktitle = {32nd DEXA International Conference}, year = {2021} }</pre>

ID	Auteur	ID	Affiliation
1	Abderrazek Azri	UL2	University of Lyon 2, France
2	Cécile Favre	UL2	University of Lyon 2, France
3	Ciprian-Octavian Truica	UPB	University Politehnica of Bucharest, Romania
		AU	Aarhus University, Denmark
4	Elena Simona Apostol	UPB	University Politehnica of Bucharest, Romania
5	Fatma Abdelhédi	TRI	Trimane company, France
6	Franck Ravat	UT1	University of Toulouse 1, France
7	Ira Assent	AU	Aarhus University, Denmark
8	Jérôme Darmont	UL2	University of Lyon 2, France
9	Nouria Harbi	UL2	University of Lyon 2, France
10	Olivier Teste	UT2	University of Toulouse 2, France
11	Pegdwendé Nicolas Sawadogo	UL2	University of Lyon 2, France
12	Yuzhao Yang	UT1	University of Toulouse 1, France

Téléchargement des données : <https://eric.univ-lyon2.fr/jdarmont/docs/m2malia-bdm-data.txt>

## Travail à effectuer

I. À partir des métadonnées à votre disposition, créer deux bases de métadonnées, l'une en JSON (adopter le mode n-uplets) et l'autre avec Neo4J. Les sauvegarder !

II. Formuler les requêtes suivantes en JSONiq et Cypher, respectivement.

1. Publications parues en 2022
2. Noms et affiliations des auteurs
3. Identifiants de publication et noms des auteurs correspondants, triés par identifiant de publication
4. Noms des revues et des auteurs qui y ont publié, triés par nom de revue
5. Noms des auteurs qui collaborent avec Yuzhao Yang
6. Nombre de publications par types, trié par type
7. Nombre d'auteurs par publications.
8. Nombre moyen d'auteurs dans les publications

III. Quelles différences voyez-vous entre JSON/RumbleDB/JSONiq et Neo4J/Cypher ? Quel outil est le plus adapté dans ce cas d'usage ?

## Correction JSON/RumbleDB/JSONiq

### // publis.json

```
{ "id" : "TruicaADA21", "type" : "article", "title" : "TextBenDS : a Generic Textual Data Benchmark for Distributed Systems", "journal" : "Information System Frontiers", "year" : 2021 }
{ "id" : "SawadogoD21a", "type" : "article", "title" : "On data lake architectures and metadata management", "journal" : "Journal of Intelligent Information Systems", "year" : 2021 }
{ "id" : "YangDRT22", "type" : "inproceedings", "title" : "Dimensional Data KNN-Based Imputation", "booktitle" : "26th ADBIS European Conference", "year" : 2022 }
{ "id" : "SawadogoD21b", "type" : "inproceedings", "title" : "Benchmarking Data Lakes Featuring Structured and Unstructured Data with DLBench", "booktitle" : "23rd DaWaK International Conference", "year" : 2021 }
{ "id" : "AzriFHDN21", "type" : "inproceedings", "title" : "Calling to CNN-LSTM for Rumor Detection : A Deep Multi-channel Model for Message Veracity Classification in Microblogs", "booktitle" : "ECML-PKDD European Conference", "year" : 2021 }
{ "id" : "YangADRT21", "type" : "inproceedings", "title" : "Internal Data Imputation in Data Warehouse Dimensions", "booktitle" : "32nd DEXA International Conference", "year" : 2021 }
```

### // authors.json

```
{ "id" : 1, "name" : "Abderrazek Azri", "affiliation" : ["University of Lyon 2, France"] }
{ "id" : 2, "name" : "Cecile Favre", "affiliation" : ["University of Lyon 2, France"] }
{ "id" : 3, "name" : "Ciprian-Octavian Truica", "affiliation" : ["University Politehnica of Bucharest, Romania", "Aarhus University, Denmark"] }
{ "id" : 4, "name" : "Elena Simona Apostol", "affiliation" : ["University Politehnica of Bucharest, Romania"] }
{ "id" : 5, "name" : "Fatma Abdelhedi", "affiliation" : ["Trimane company, France"] }
{ "id" : 6, "name" : "Franck Ravat", "affiliation" : ["University of Toulouse 1, France"] }
{ "id" : 7, "name" : "Ira Assent", "affiliation" : ["Aarhus University, Denmark"] }
{ "id" : 8, "name" : "Jerome Darmont", "affiliation" : ["University of Lyon 2, France"] }
{ "id" : 9, "name" : "Nouria Harbi", "affiliation" : ["University of Lyon 2, France"] }
{ "id" : 10, "name" : "Olivier Teste", "affiliation" : ["University of Toulouse 2, France"] }
{ "id" : 11, "name" : "Pegdwende Nicolas Sawadogo", "affiliation" : ["University of Lyon 2, France"] }
{ "id" : 12, "name" : "Yuzhao Yang", "affiliation" : ["University of Toulouse 1, France"] }
```

### // pubauth.json

```
{ "idpub" : "TruicaADA21", "idauth" : 3 }
{ "idpub" : "TruicaADA21", "idauth" : 4 }
{ "idpub" : "TruicaADA21", "idauth" : 8 }
{ "idpub" : "TruicaADA21", "idauth" : 7 }
{ "idpub" : "SawadogoD21a", "idauth" : 11 }
{ "idpub" : "SawadogoD21a", "idauth" : 8 }
{ "idpub" : "YangDRT22", "idauth" : 12 }
{ "idpub" : "YangDRT22", "idauth" : 8 }
{ "idpub" : "YangDRT22", "idauth" : 6 }
{ "idpub" : "YangDRT22", "idauth" : 10 }
{ "idpub" : "SawadogoD21b", "idauth" : 11 }
{ "idpub" : "SawadogoD21b", "idauth" : 8 }
{ "idpub" : "AzriFHDN21", "idauth" : 1 }
{ "idpub" : "AzriFHDN21", "idauth" : 2 }
{ "idpub" : "AzriFHDN21", "idauth" : 9 }
{ "idpub" : "AzriFHDN21", "idauth" : 8 }
{ "idpub" : "YangADRT21", "idauth" : 12 }
{ "idpub" : "YangADRT21", "idauth" : 5 }
{ "idpub" : "YangADRT21", "idauth" : 8 }
{ "idpub" : "YangADRT21", "idauth" : 6 }
{ "idpub" : "YangADRT21", "idauth" : 10 }
```

### ( : RQ1 :)

```
for $p in json-file("publis.json")
where $p.year eq 2022
return $p
```

**(: RQ2 :)**

```
for $a in json-file("authors.json")
return { $a.name : $a.affiliation }
```

**(: RQ3 :)**

```
for $b in json-file("pubauth.json"),
  $a in json-file("authors.json")
where $b.idauth eq $a.id
order by $b.idpub
return { $b.idpub : $a.name }
```

**(: RQ4 :)**

```
for $p in json-file("publis.json"),
  $b in json-file("pubauth.json"),
  $a in json-file("authors.json")
where $p.type eq "article"
  and $p.id eq $b.idpub
  and $b.idauth eq $a.id
order by $p.journal
return { $p.journal : $a.name }
```

**(: RQ5 :)**

```
let $datn := [
  for $a1 in json-file("authors.json"),
    $b1 in json-file("pubauth.json"),
    $b2 in json-file("pubauth.json"),
    $a2 in json-file("authors.json")
  where $a1.name eq "Yuzhao Yang"
    and $a1.id eq $b1.idauth
    and $b2.idauth eq $a2.id
    and $b1.idpub eq $b2.idpub
    and $a2.name ne "Yuzhao Yang"
  return $a2.name ]
return [ distinct-values($data[]) ]
```

**(: RQ6 :)**

```
for $p in json-file("publis.json")
let $type := $p.type
group by $type
order by $type
return { $type : count($p.id) }
```

**(: RQ7 :)**

```
for $p in json-file("publis.json"),
  $b in json-file("pubauth.json")
where $p.id eq $b.idpub
let $publi := $p.id
group by $publi
return { $publi : count($p.id) }
```

**(: RQ8 :)**

```
let $comptes := [
  for $p in json-file("publis.json"),
    $b in json-file("pubauth.json")
  where $p.id eq $b.idpub
  let $publi := $p.id
  group by $publi
  return count($p.id) ]
return { "moyenne" : avg($comptes[]) }
```

## Correction Neo4J/Cypher

### // Nœuds publications

```
create(p : pub {id : "TruicaADA21", type : "article", title : "TextBenDS: a Generic Textual Data Benchmark for Distributed Systems", journal : "Information System Frontiers", year : 2021})
create(p : pub {id : "SawadogoD21a", type : "article", title : "On data lake architectures and metadata management", journal : "Journal of Intelligent Information Systems", year : 2021})
create(p : pub {id : "YangDRT22", type : "inproceedings", title : "Dimensional Data KNN-Based Imputation", booktitle : "26th ADBIS European Conference", year : 2022})
create(p : pub {id : "SawadogoD21b", type : "inproceedings", title : "Benchmarking Data Lakes Featuring Structured and Unstructured Data with DLBench", booktitle : "23rd DaWaK International Conference", year : 2021})
create(p : pub {id : "AzriFHDN21", type : "inproceedings", title : "Calling to CNN-LSTM for Rumor Detection: A Deep Multi-channel Model for Message Veracity Classification in Microblogs", booktitle : "ECML-PKDD European Conference", year : 2021})
create(p : pub {id : "YangADRT21", type : "inproceedings", title : "Internal Data Imputation in Data Warehouse Dimensions", booktitle : "32nd DEXA International Conference", year : 2021})
```

### // Nœuds auteurs

```
create(a : author {id : 1, name : "Abderrazek Azri"})
create(a : author {id : 2, name : "Cecile Favre"})
create(a : author {id : 3, name : "Ciprian-Octavian Truica"})
create(a : author {id : 4, name : "Elena Simona Apostol"})
create(a : author {id : 5, name : "Fatma Abdelhedi"})
create(a : author {id : 6, name : "Franck Ravat"})
create(a : author {id : 7, name : "Ira Assent"})
create(a : author {id : 8, name : "Jerome Darmon"})
create(a : author {id : 9, name : "Nouria Harbi"})
create(a : author {id : 10, name : "Olivier Teste"})
create(a : author {id : 11, name : "Pegdwende Nicolas Sawadogo"})
```

### // Nœuds affiliations

```
create(f : affiliation {id : "UL2", affiliation : "University of Lyon 2, France"})
create(f : affiliation {id : "UPB", affiliation : "University Politehnica of Bucharest, Romania"})
create(f : affiliation {id : "AU", affiliation : "Aarhus University, Denmark"})
create(f : affiliation {id : "TRI", affiliation : "Trimane company, France"})
create(f : affiliation {id : "UT1", affiliation : "University of Toulouse 1, France"})
create(f : affiliation {id : "UT2", affiliation : "University of Toulouse 2, France"})
```

### // Relations bilatérales publications-auteurs

```
match (p : pub), (a : author) where p.id = "TruicaADA21" and a.id = 3 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "TruicaADA21" and a.id = 4 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "TruicaADA21" and a.id = 8 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "TruicaADA21" and a.id = 7 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "TruicaADA21" and a.id = 3 create (a)-[:authorOf]->(p)
match (p : pub), (a : author) where p.id = "TruicaADA21" and a.id = 4 create (a)-[:authorOf]->(p)
match (p : pub), (a : author) where p.id = "TruicaADA21" and a.id = 8 create (a)-[:authorOf]->(p)
match (p : pub), (a : author) where p.id = "TruicaADA21" and a.id = 7 create (a)-[:authorOf]->(p)

match (p : pub), (a : author) where p.id = "SawadogoD21a" and a.id = 11 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "SawadogoD21a" and a.id = 8 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "SawadogoD21a" and a.id = 11 create (a)-[:authorOf]->(p)
match (p : pub), (a : author) where p.id = "SawadogoD21a" and a.id = 8 create (a)-[:authorOf]->(p)

match (p : pub), (a : author) where p.id = "YangDRT22" and a.id = 12 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "YangDRT22" and a.id = 8 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "YangDRT22" and a.id = 6 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "YangDRT22" and a.id = 10 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "YangDRT22" and a.id = 12 create (a)-[:authorOf]->(p)
match (p : pub), (a : author) where p.id = "YangDRT22" and a.id = 8 create (a)-[:authorOf]->(p)
match (p : pub), (a : author) where p.id = "YangDRT22" and a.id = 6 create (a)-[:authorOf]->(p)
```

```

match (p : pub), (a : author) where p.id = "YangDRT22" and a.id = 10 create (a)-[:authorOf]->(p)

match (p : pub), (a : author) where p.id = "SawadogoD21b" and a.id = 11 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "SawadogoD21b" and a.id = 8 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "SawadogoD21b" and a.id = 11 create (a)-[:authorOf]->(p)
match (p : pub), (a : author) where p.id = "SawadogoD21b" and a.id = 8 create (a)-[:authorOf]->(p)

match (p : pub), (a : author) where p.id = "AzriFHdn21" and a.id = 1 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "AzriFHdn21" and a.id = 2 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "AzriFHdn21" and a.id = 9 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "AzriFHdn21" and a.id = 8 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "AzriFHdn21" and a.id = 1 create (a)-[:authorOf]->(p)
match (p : pub), (a : author) where p.id = "AzriFHdn21" and a.id = 2 create (a)-[:authorOf]->(p)
match (p : pub), (a : author) where p.id = "AzriFHdn21" and a.id = 9 create (a)-[:authorOf]->(p)
match (p : pub), (a : author) where p.id = "AzriFHdn21" and a.id = 8 create (a)-[:authorOf]->(p)

match (p : pub), (a : author) where p.id = "YangADRT21" and a.id = 12 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "YangADRT21" and a.id = 5 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "YangADRT212" and a.id = 8 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "YangADRT21" and a.id = 6 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "YangADRT21" and a.id = 10 create (p)-[:authoredBy]->(a)
match (p : pub), (a : author) where p.id = "YangADRT21" and a.id = 12 create (a)-[:authorOf]->(p)
match (p : pub), (a : author) where p.id = "YangADRT21" and a.id = 5 create (a)-[:authorOf]->(p)
match (p : pub), (a : author) where p.id = "YangADRT21" and a.id = 8 create (a)-[:authorOf]->(p)
match (p : pub), (a : author) where p.id = "YangADRT21" and a.id = 6 create (a)-[:authorOf]->(p)
match (p : pub), (a : author) where p.id = "YangADRT21" and a.id = 10 create (a)-[:authorOf]->(p)

```

#### // Relations auteurs-affiliations

```

match (a : author), (f : affiliation) where a.id = 1 and f.id = "UL2" create (a)-[:isAffiliate]->(f)
match (a : author), (f : affiliation) where a.id = 2 and f.id = "UL2" create (a)-[:isAffiliate]->(f)
match (a : author), (f : affiliation) where a.id = 3 and f.id = "UPB" create (a)-[:isAffiliate]->(f)
match (a : author), (f : affiliation) where a.id = 3 and f.id = "AU" create (a)-[:isAffiliate]->(f)
match (a : author), (f : affiliation) where a.id = 4 and f.id = "UPB" create (a)-[:isAffiliate]->(f)
match (a : author), (f : affiliation) where a.id = 5 and f.id = "TRI" create (a)-[:isAffiliate]->(f)
match (a : author), (f : affiliation) where a.id = 6 and f.id = "UT1" create (a)-[:isAffiliate]->(f)
match (a : author), (f : affiliation) where a.id = 7 and f.id = "AU" create (a)-[:isAffiliate]->(f)
match (a : author), (f : affiliation) where a.id = 8 and f.id = "UL2" create (a)-[:isAffiliate]->(f)
match (a : author), (f : affiliation) where a.id = 9 and f.id = "UL2" create (a)-[:isAffiliate]->(f)
match (a : author), (f : affiliation) where a.id = 10 and f.id = "UT2" create (a)-[:isAffiliate]->(f)
match (a : author), (f : affiliation) where a.id = 11 and f.id = "UL2" create (a)-[:isAffiliate]->(f)
match (a : author), (f : affiliation) where a.id = 12 and f.id = "UT1" create (a)-[:isAffiliate]->(f)

```

#### // RQ1

```

match(p : pub {year : 2022})
return(p)

```

#### // RQ2

```

match(a : author)-[:isAffiliate]-(f : affiliation)
return a.name, f.affiliation

```

#### // RQ3

```

match(p : pub)-[:authoredBy]-(a : author)
return p.id, a.name
order by p.id

```

#### // RQ4

```

match(p : pub)-[:authoredBy]-(a : author)
where p.type = "article"
return p.journal, a.name
order by p.journal

```

**// RQ5**

```
match(a1 : author)-[:authorOf]-(p : pub)-[:authoredBy]-(a2 : author)
where a1.name ="Yuzhao Yang" and a2.name <> "Yuzhao Yang"
return distinct a2.name
```

**// RQ6**

```
match(p : pub)
return p.type, count(p)
order by p.type
```

**// RQ7**

```
match(p : pub)-[:authoredBy]-(a : author)
return p.id, count(a)
```

**// RQ8**

```
match(p : pub)-[:authoredBy]-(a : author)
with p.id as i, count(a) as c
return avg(c)
```

**Correction conclusion**

- Les données de JSON, sous leur forme de n-uplets, sont plus simples à mettre en œuvre que les nœuds et les relations exprimées en Cypher.
- Les requêtes d'interrogation Cypher sont plus compactes/simples que les requêtes JSONiq.
- Des comparaisons de performance (temps de réponse des requêtes, passage à l'échelle...) devraient être envisagées pour trancher en toute objectivité !