Introduction to Databases

Lecture 7 – Graph databases: Neo4j

Gianluca Quercini

gianluca.quercini@centralesupelec.fr

Master DSBA 2020 – 2021

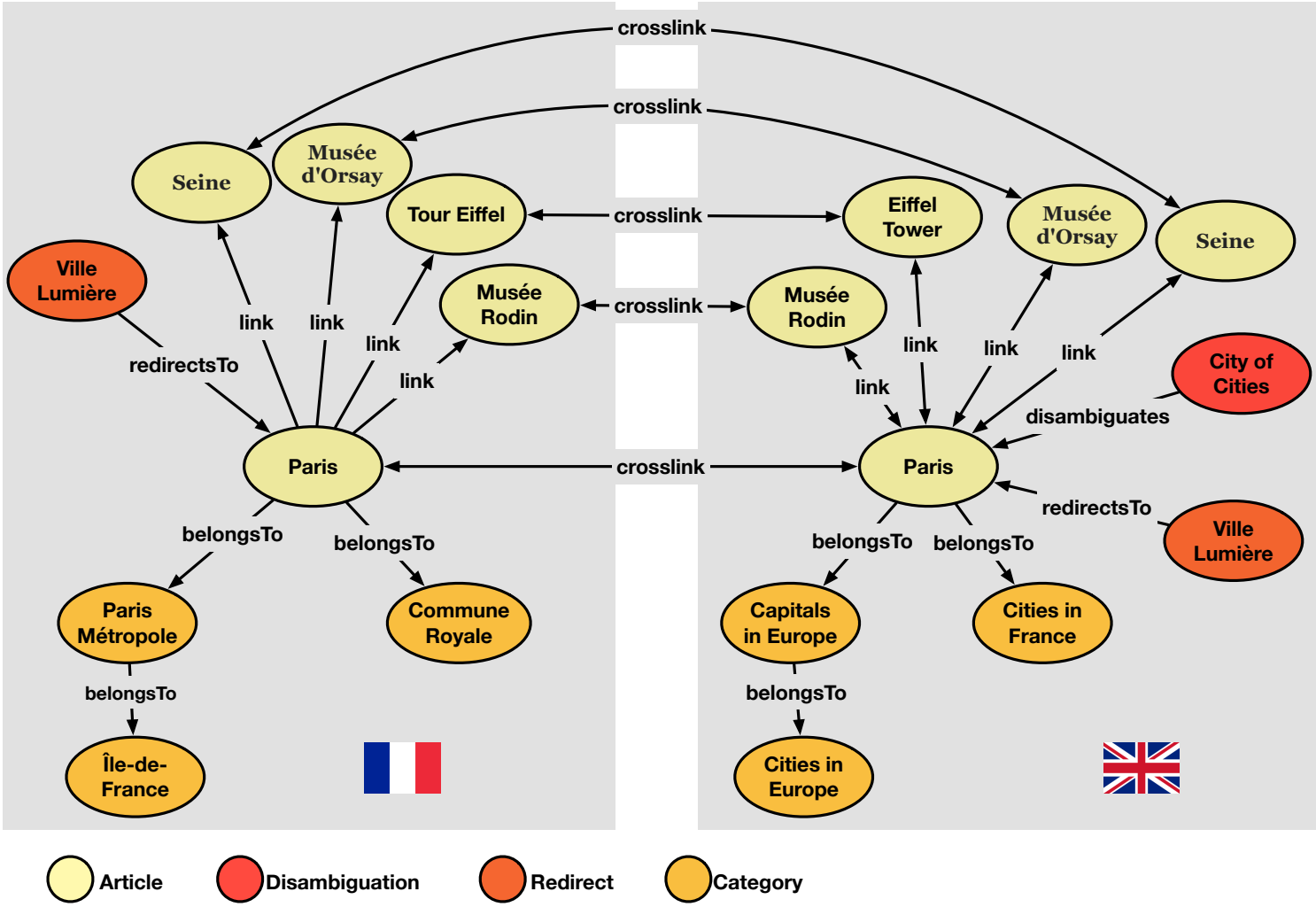
CentraleSupélec

What you will learn

In this lecture you will learn:

- What a graph database is.
- Why relational and aggregate-based NoSQL databases are not a good choice to represent large graphs.
- The basic concepts of Neo4j.
- Neo4j query language: Cypher.

An example: Wikipedia



Representing a graph in a relational database

- Impedance mismatch. Graph represented as a set of tables.

Page			
id	title	lang	type
0	Paris	en	article
1	Musée Rodin	en	article
2	Eiffel Tower	en	article
.	.	.	.
.	.	.	.
.	.	.	.
50	Tour Eiffel	fr	article
.	.	.	.
.	.	.	.
.	.	.	.
99	Capitals in Europe	en	category
.	.	.	.
.	.	.	.

Link		
src	dst	type
0	1	link
0	2	link
0	99	belongsTo
1	0	link
2	0	link
.	.	.
.	.	.
.	.	.
2	50	crosslink
.	.	.
.	.	.

Graph traversal in SQL

- n: number of pages.
- Indexes (B-trees) on `Page.id`, `Page.title`, `Page.lang`, `Link.src`.

Get the articles that have a link from `Paris`

```
SELECT p1.title
FROM Page p1 JOIN Link l ON p1.id = l.dst
  JOIN Page p2 ON p2.id = l.src
WHERE p2.title = 'Paris' AND p2.lang = 'en'
```

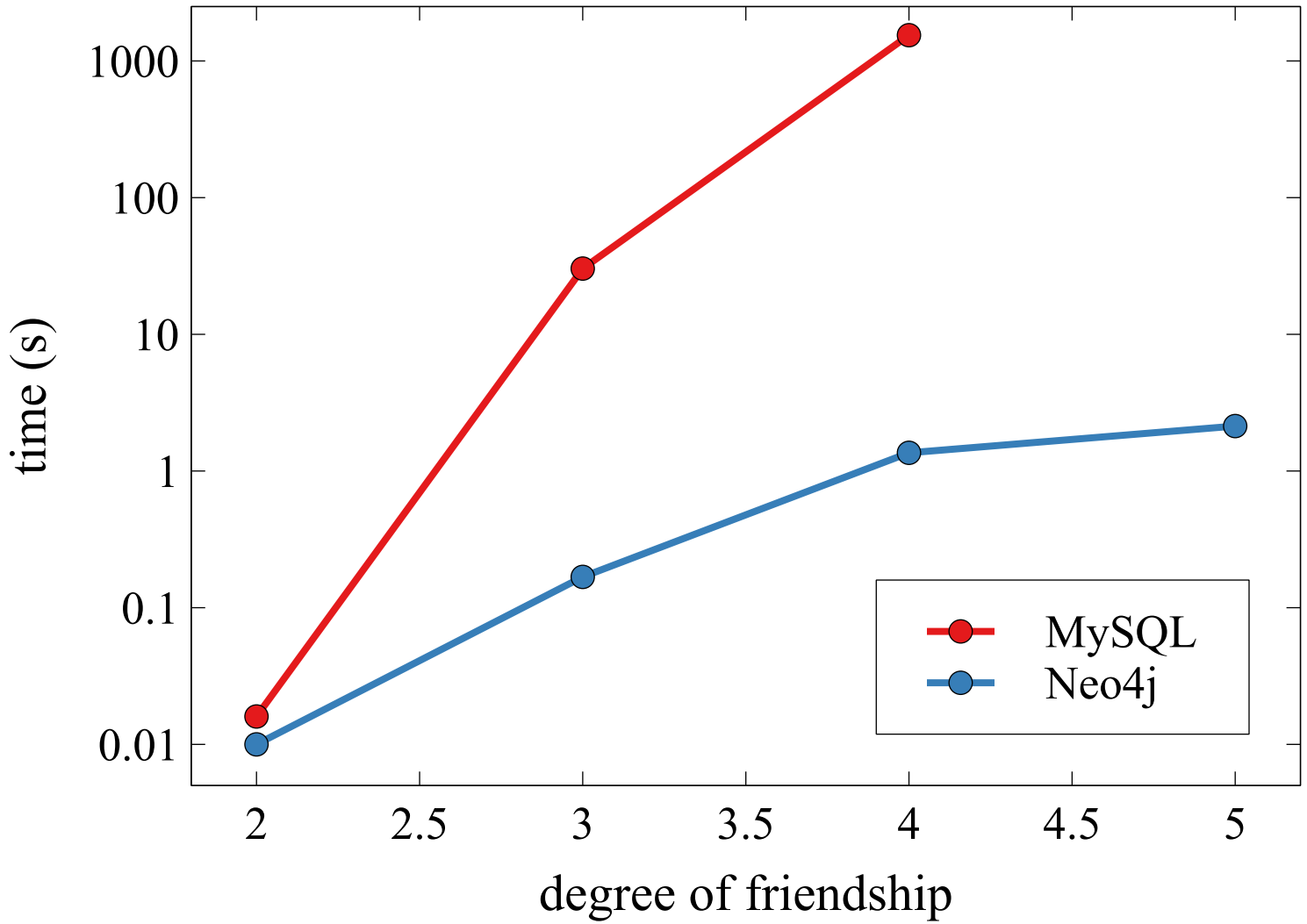
- Two join operations: `Page p1` $\bowtie$ (`Link l` $\bowtie$ `Page p2`)
  - `Link l` $\bowtie$ `Page p2` $=$ `Temp` with κ rows.
- Cost of WHERE: $O(\log n)$.
- Cost of `Link l` $\bowtie$ `Page p2`: $O(\log n) + \kappa$.
- Cost of `Page p` $\bowtie$ `Temp`: $\kappa \cdot O(\log n)$.
- The cost of the query depends on the size of the whole graph.
  - Even if the operation is local to a node.

Graph traversal in SQL

Articles that have a link from the articles that have a link from Paris

```
SELECT p2.title
FROM Page p1 JOIN Link l1
    ON p1.id = l.src
  JOIN Link l2
    ON l2.src = l1.dst
  JOIN Page p2
    ON l2.dst = p2.id
WHERE p1.title = 'Paris' AND p1.lang = 'en'
AND l2.dst <> p1.id
```

Graph databases Vs Relational databases



Graphs in aggregate NoSQL databases

```
{
    nodeId : 0,
    title    : "Paris",
    lang     : "en",
    type     : "article",
    linksTo :  [
            {
                edgeId : 0,
                src      : 0,
                dst      : 1,
                type     : "link"
            },
            {
                edgeId : 1,
                src      : 0
                dst      : 2,
                type     : "link"
            },
            ........
        ]
}
```

```
{
    nodeId : 1,
    title    : "Musée Rodin",
    lang     : "en",
    type     : "article",
    linksTo :  [
            ....
        ]
}

{
    nodeId : 2,
    title    : "Eiffel Tower",
    lang     : "en",
    type     : "article",
    linksTo :  [
            ....
        ]
}
```

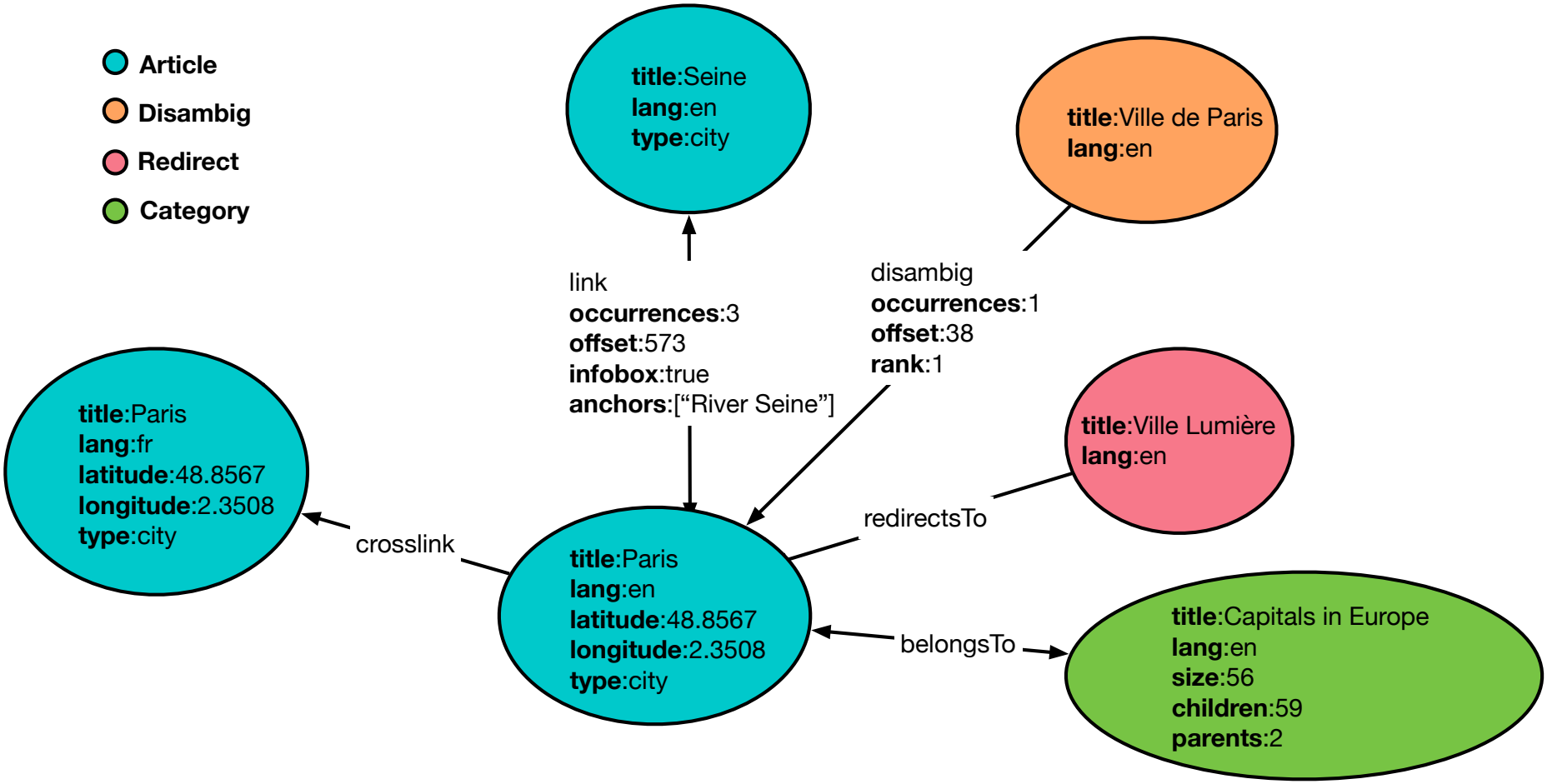
Graphs in relational and NoSQL databases

In relational and aggregate NoSQL databases relationships are not explicitly represented. This is why they're inefficient on graph traversal operations.

Neo4j

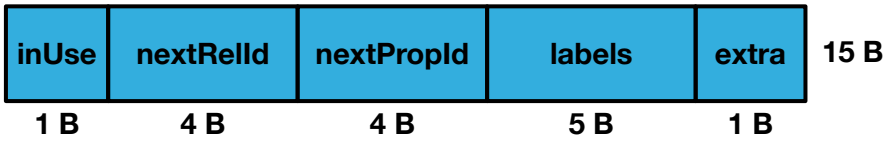
- Neo4j is the most used graph database today.
- Neo4j uses the labelled property graph model to represent a graph.
- Neo4j provides a query language called **Cypher**.
  - declarative language (like SQL).
  - not standard (unlike SQL).
- Neo4j supports a standard language: SPARQL
  - Used in the Semantic Web to query RDF data.
  - Syntax similar to SQL.
  - Complicated syntax.

The labelled property graph model

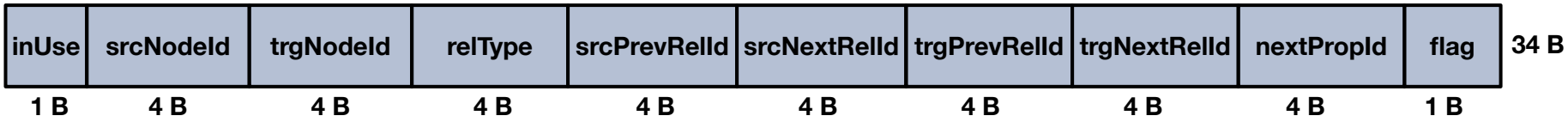


★ Storage in Neo4j

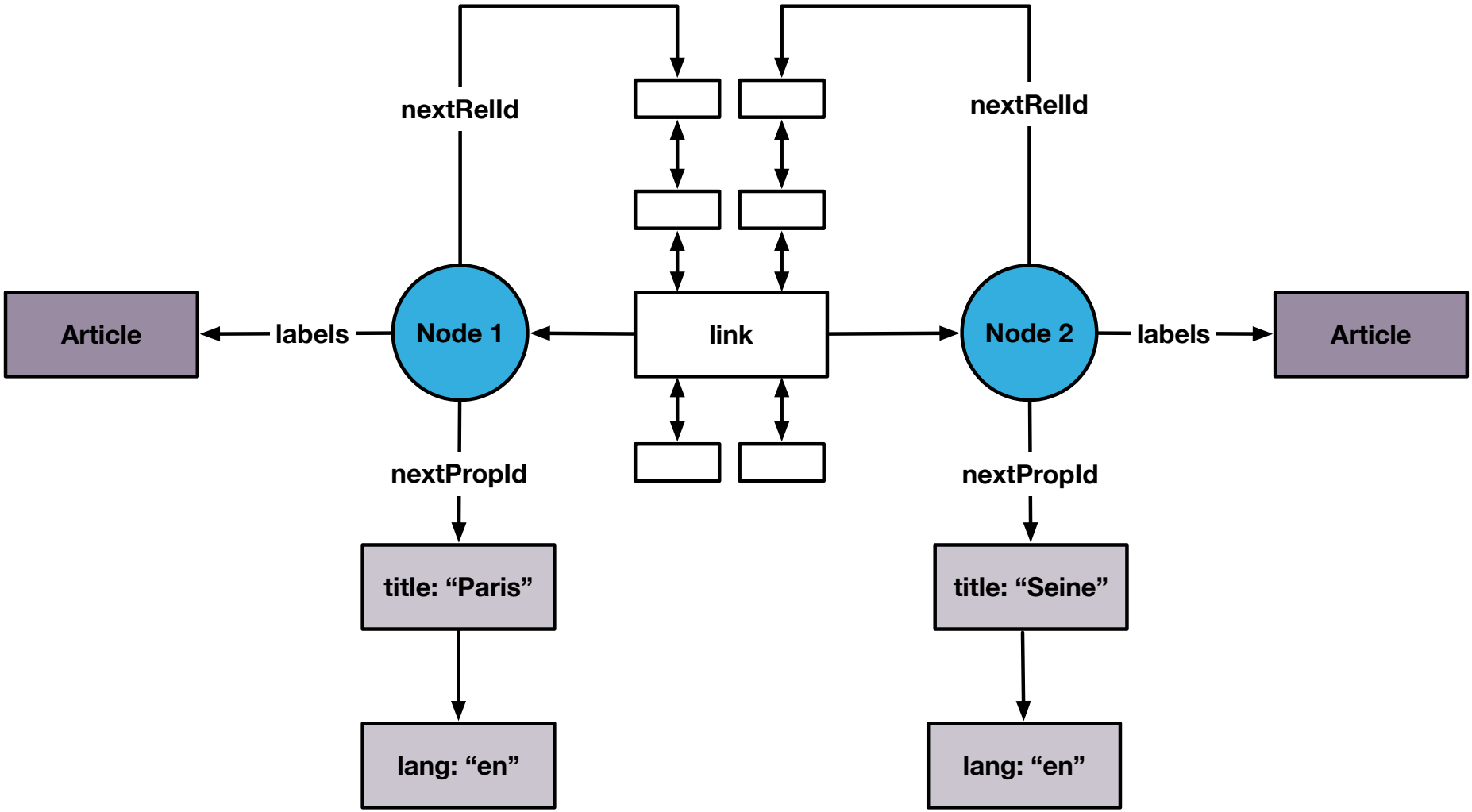
Node record



Relationship record



★ Storage in Neo4j

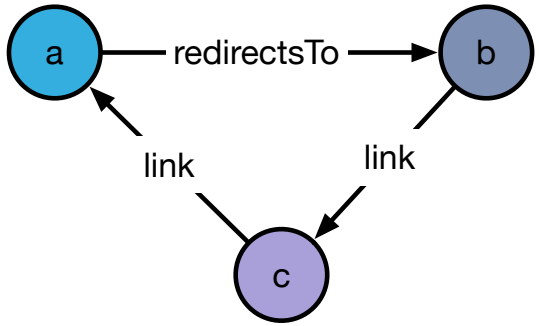


Patterns in Cypher

- Idea of Cypher: query by drawing patterns.

```
(a)-[:redirectsTo]->(b)
```


```
(a)-[:redirectsTo]->(b)-[:link]->(c)-[:link]->(a)
```



Binding a pattern to specific nodes and links

```
(a:Redirect)-[:redirectsTo]->(b:Article {title:'Paris', lang:'en'})
```

Queries

```
MATCH  (n:Article {title:"Paris"})
RETURN n
```

```
MATCH (n:Article)
WHERE n.title="Paris"
RETURN n
```

```
MATCH (:Article {title:"Paris", lang:"en"})-[:link]->(m:Article)
RETURN m.title
```

```
MATCH (n:Article)
RETURN DISTINCT n.title
```

```
MATCH (n:Article {title:"Paris", lang:"en"})-[r:link]->(m:Article)
RETURN *
```

Aggregating functions in Cypher

```
MATCH (n:Article {title:"Paris", lang:"en"})-[:belongsTo]->(m:Category)
RETURN n.title, COUNT(m) AS nbCategories
```

```
MATCH (n:Article)-[r:link]->(m:Article)
RETURN n.title, COUNT(r) AS nbLinks
ORDER BY nbLinks DESC
```

```
MATCH (n:Article)-[r:link]->(m:Article)
WITH n.title AS title, COUNT(r) AS nbLinks
RETURN AVG(nbLinks)
```

★ Transactions

- Since its inception, Neo4j has supported ACID transactions.
- It was an exception in the NoSQL world.
- The primary goal of a graph database was not data distribution.
- Today other NoSQL databases acknowledge the usefulness of ACID transactions (even in a distributed context).