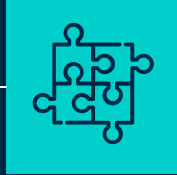




Chen Zhihao, Maxence Hennekein, Baptiste Petrau, Loïc Dumay

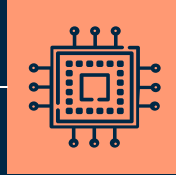
ESiEE[it]
L'école de l'expertise numérique

SOMMAIRE



01

ORIGIN &
UTILISATION



02

ARCHITECTURE



03

DÉMONSTRATION

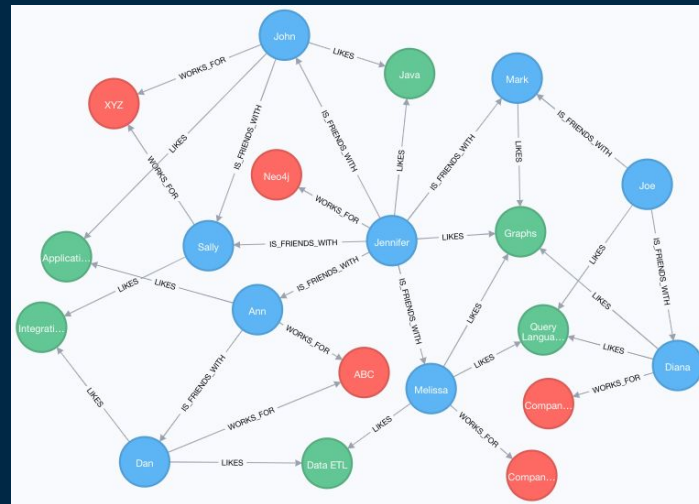
01

ORIGIN & UTILISATION

1.1 ORIGIN

1. Qu'est-ce que neo4j ?

- Un DBMS (ou SGBD en français) basé sur les graphes
- Développé en Java et OpenSource
- Il possède des caractéristiques principales :
 - Transaction
 - Haute disponibilité
 - Volumétrique
 - Schemaless
- Son langage est le Cypher :
 - un langage de requête graphe déclaratif
- Créer en 2001
 - Version 1.0 : février 2010
 - Dernière version 4.3.6 : octobre 2021
- Site officiel : <https://neo4j.com/>



But : permettre l'analyse d'un grand nombre de données

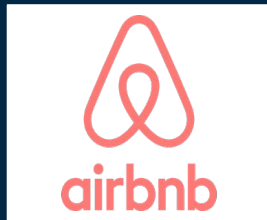
Gains : rapidité et sans limite

1.2 UTILISATION

Finance	Télécom	Santé	RH	Média	Web Social	Industrie & Logistique
UBS First Data. ADP ADVENT NOMURA veda die Bayerische ICE	CISCO hp SFR EarthLink telenor maaii EarthLink	GoodStart Janssen ZEPHYR curaspan doximity Care.com	careerbuilder InfoJobs EQUILAR SRM SOCIETY FOR HUMAN RESOURCE MANAGEMENT	NATIONAL GEOGRAPHIC LIFECHURCH.TV codex Lifestation Perigee TechCrunch	classmates eHarmony Hinge me9tic SNAP INTERACTIVE	ebay now noble group SNCF KiwiRail ConocoPhillips Impact Technologies
Jeux	Commerce	Business Services	Information Services			
Bally bwin.party 4 gamesys YELAGO	Walmart cobrain Juice PLUS+	compete Cerved Group scribestar research now	LOCKHEED MARTIN Lufthansa Systems			

Quelques entreprises qui ont déjà adopté Neo4j.

1.2 UTILISATION



Gestion des données de
référence

Objectif :

Créer un moteur de recherche de
ressources de données simple et fiables.



Traitement du langage
naturel

Objectif :

Créer un outil de TLN (traitement de
langage naturel) pour l'extraction à
grande échelle du sens de documents.



Gestion de la nomenclature et de
la chaîne d'approvisionnement

Objectif :

Moderniser les processus
d'approvisionnement, de budgétisation
et de logistique de l'Armée Américaine

Source d'informations :

AirBnB : <https://go.neo4j.com/rs/710-RRR-335/images/Neo4j-case-study-Airbnb-FR.pdf>
Caterpillar : <https://go.neo4j.com/rs/710-RRR-335/images/Neo4j-case-study-Caterpillar-FR.pdf>
US Army : <https://go.neo4j.com/rs/710-RRR-335/images/Neo4j-case-study-US-army-FR.pdf>

1.2 UTILISATION

Principalement utilisé pour :

- Détecter la fraude
- Améliorer l'intelligence artificielle
- Gérer les chaîne d'approvisionnement
- Obtenir une vue à 360° des données
- La gestion de l'entretien de divers matériels
- La gestion de références
- Le traitement du langage naturel
- Gestion de la nomenclature et de la chaîne d'approvisionnement
- Et plus encore ...

ARCHITECTURE

02

2.1 ARCHITECTURE

L'architecture interne

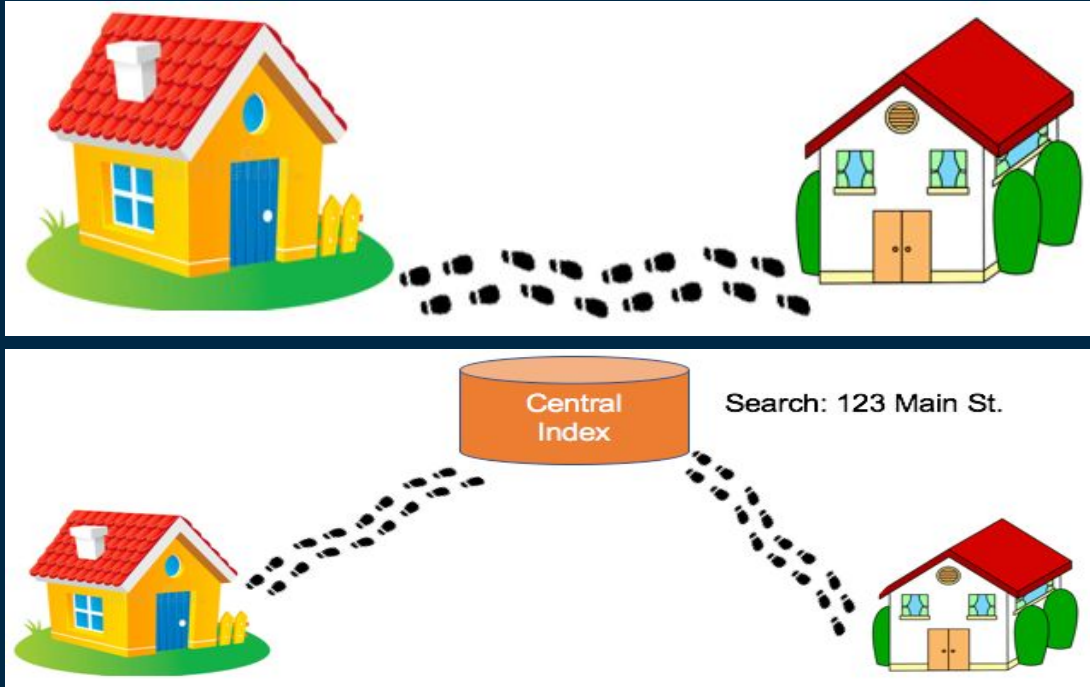
- Deux composants clés : référentiel de stockage & moteur de traitement
- Neo4j utilise son propre référentiel
- Principe de contiguité sans index
- Cypher

Les Aspects supportés par Neo4j :

- Réplication
- Architecture Maître/esclave
- Data Sharding : Fabric
- Clusters
- Multi-data Center
- Rest API

2.1 ARCHITECTURE

La continuité sans Index



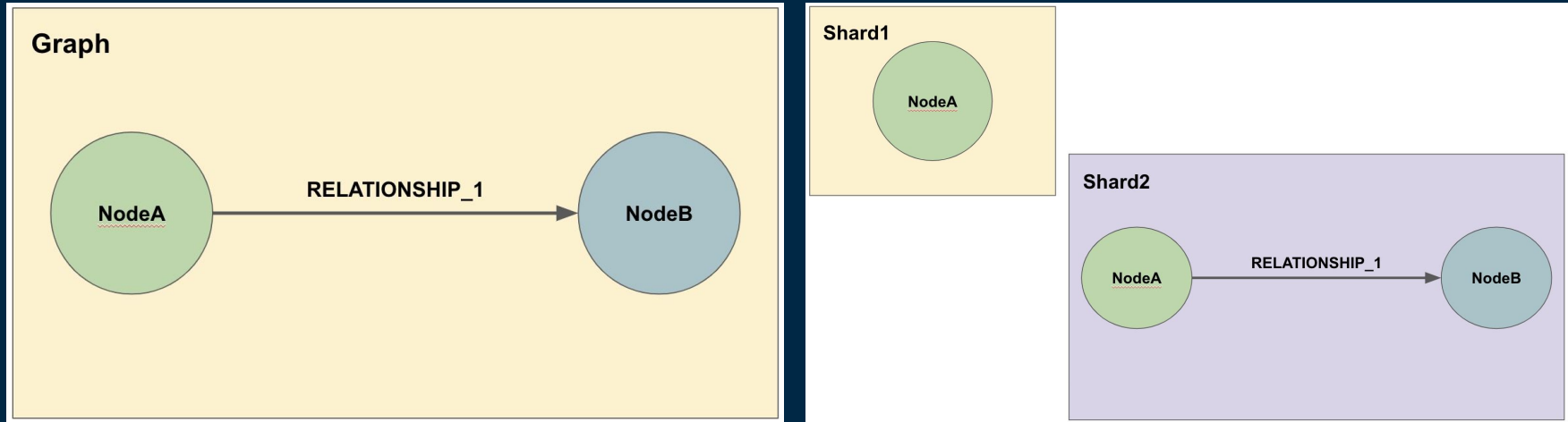
2.1 ARCHITECTURE interne

Replication

- Replication for Backup
- Replication for Data Centers
- Replica Writes
- Replica reads
- Read Repair

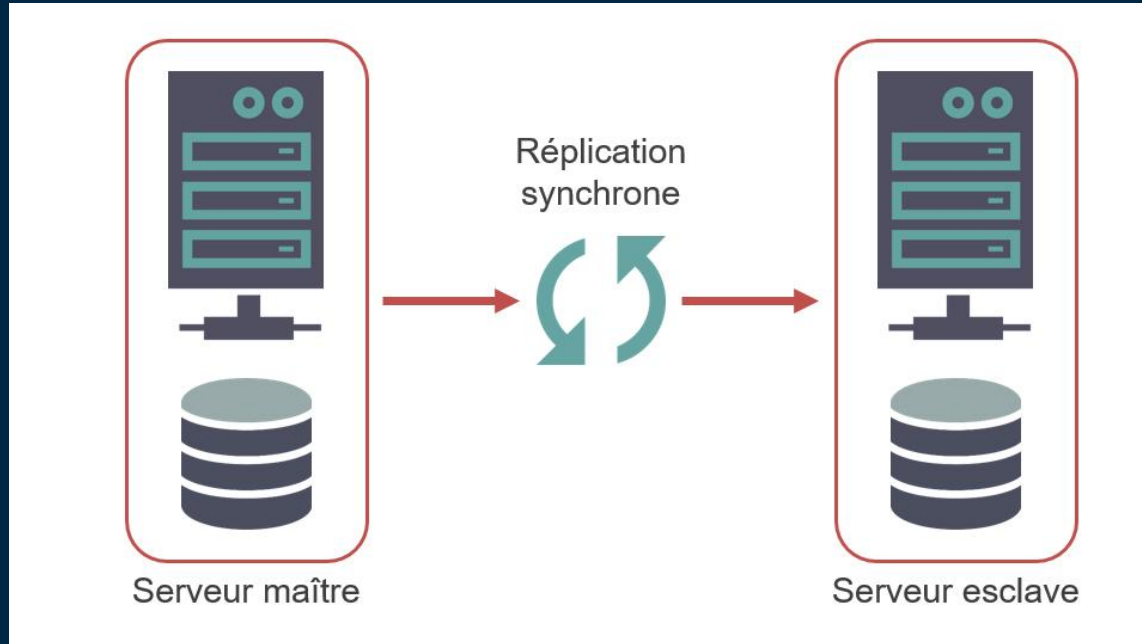
2.1 ARCHITECTURE

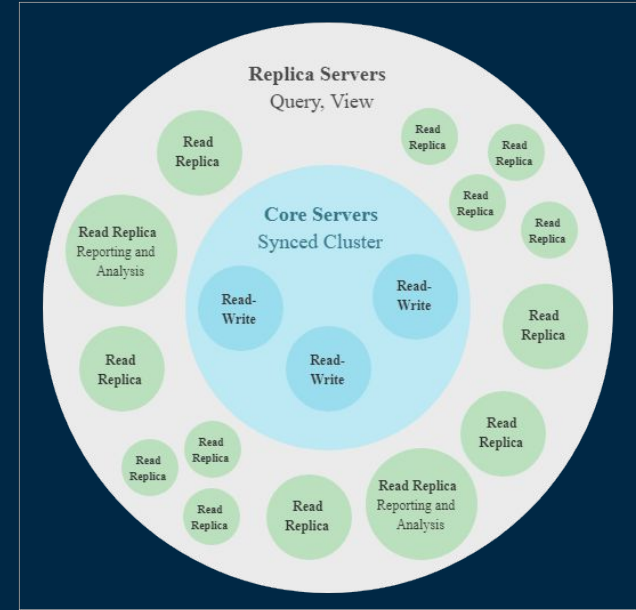
Le Data Sharding



2.1 ARCHITECTURE

La relation Maître/Esclave





2.1 ARCHITECTURE

REST API

>

http://192.168.1.106:7474/user/neo4j

⋮

☒ GET
 ☐ POST
 ☐ PUT
 ☐ PATCH
 ☐ DELETE
 ☐ HEAD
 ☐ OPTIONS

☐ Other

Raw

Form

Headers

Authorization

Basic bmVvNGo6cGFzc3dvcmQ=

✕

[ADD NEW HEADER](#)

Clear

Send

Status

200 OK

ⓘ

Loading time: 16 ms

Request headers

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.106 Safari/537.36
Authorization: Basic bmVvNGo6cGFzc3dvcmQ=
Accept: */*
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8

Response headers

Date: Wed, 09 Dec 2015 15:39:47 GMT
Content-Type: application/json; charset=UTF-8
Access-Control-Allow-Origin: *
Content-Length: 135
Server: Jetty(9.2.4.v20141103)

Raw

JSON

Response

[COPY TO CLIPBOARD](#)
[SAVE AS FILE](#)

```

{
  "password_change_required": false
  "password_change": "http://192.168.1.106:7474/user/neo4j/password"
  "username": "neo4j"
}

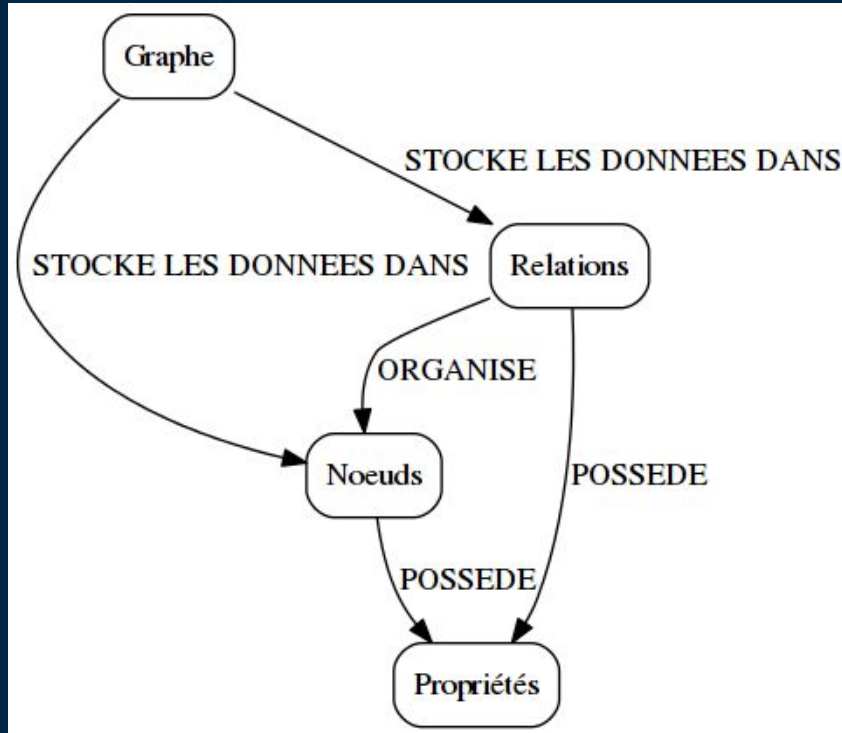
```

2.2 Modélisation des données

- BDD ORIENTÉE GRAPHES
- Questions à vous poser avant de partir sur une base de données orientée graphe :
 - Vos données sont-elles dynamiques ?
 - Vos données sont-elles connectées ?
 - Avez-vous besoin d'un schéma flexible ?
 - Devez-vous faire du temps réel ?

2.2 Modélisation des données

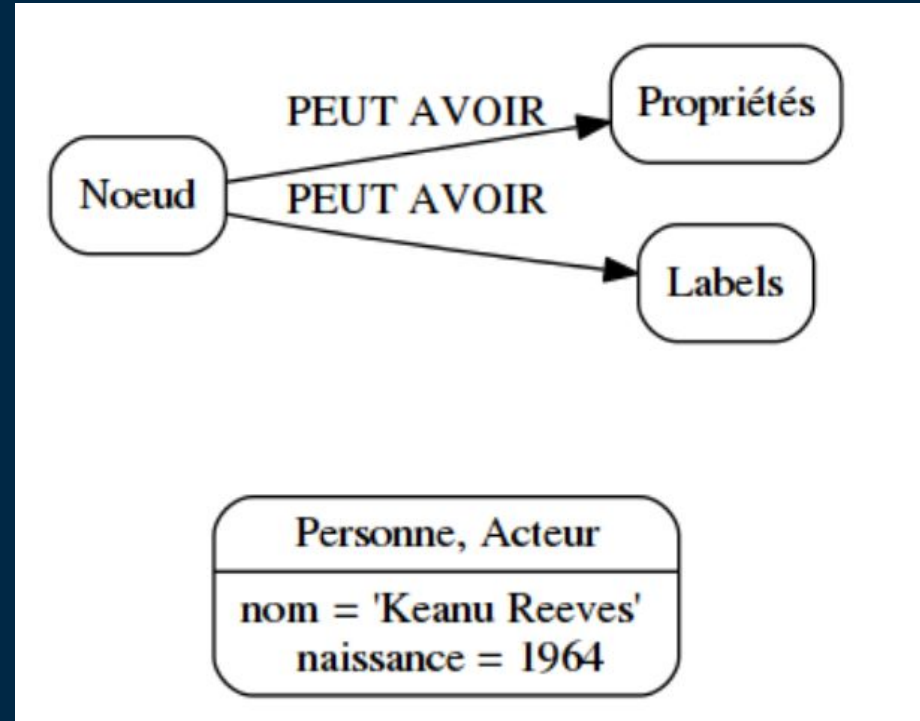
Le fonctionnement de Neo4j



2.2 Modélisation des données

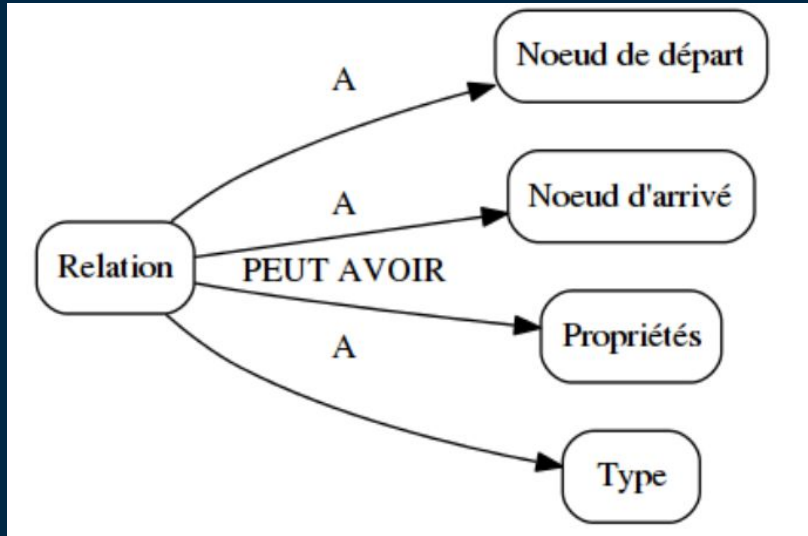
Qu'est ce que un noeud

L'unité fondamentale qui
forme un graphe est le nœud

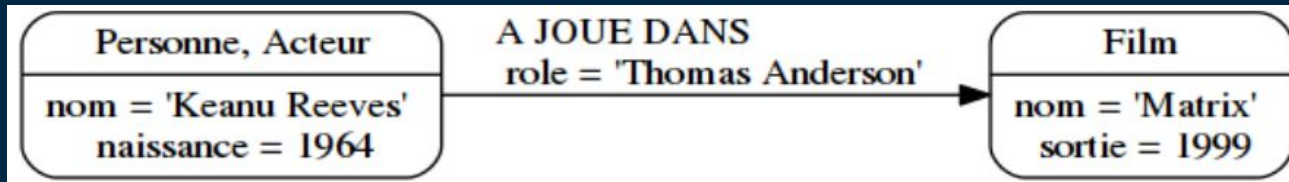


2.2 Modélisation des données

Qu'est ce qu'une relation



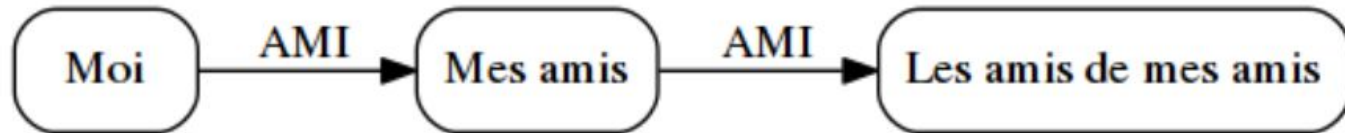
Les relations entre les nœuds sont la clef de voûte des graphes, c'est ce qui permet de lier des données et de créer des structures comme des listes, des arbres, des maps...



2.2 Modélisation des données

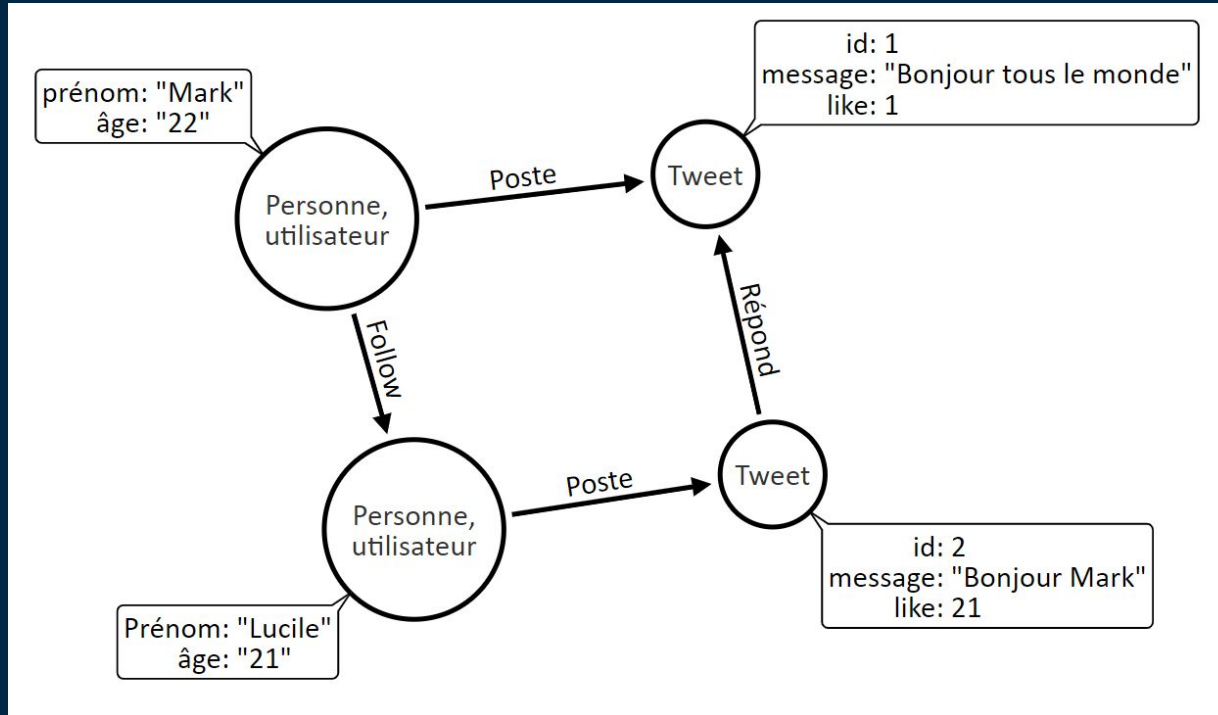
Le langage **cypher** propre à Neo4j

- **Cypher est un langage déclaratif** permettant de requêter et mettre à jour le graphe. **Inspiré du SQL**
- Son objectif est de **permettre à l'utilisateur de définir des motifs**, qui seront par la suite recherchés dans tout le graphe.



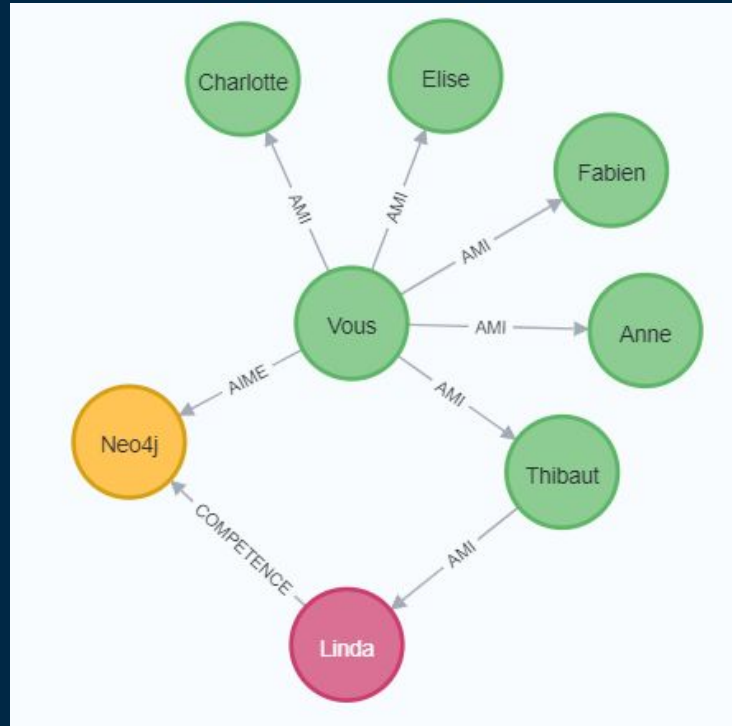
2.2 Modélisation des données

Exemple d'une structure Neo4j



2.2 Modélisation des données

Exemple d'une structure Neo4j



DÉMONSTRATION

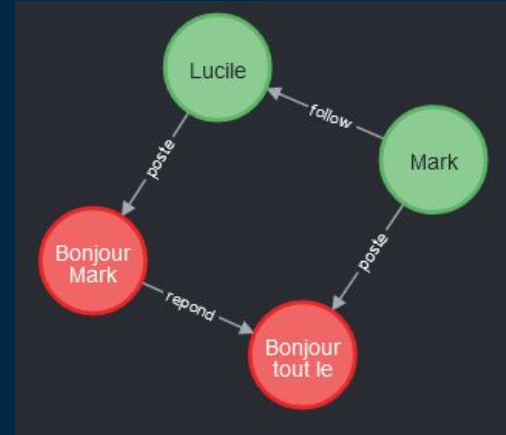
03

3. Démonstration

CRUD (Create, Read, Update, Delete)

CREATE

```
(m:Personne:utilisateur {prenom:"Mark", age:42}),  
(mess:Tweet {message:"Bonjour tout le monde", like:1}),  
(l:Personne:utilisateur {prenom:"Lucile", age:21}),  
(rep:Tweet {message:"Bonjour Mark", like:21}),  
(m)-[:poste]→(mess),  
(m)-[:follow]→(l),  
(l)-[:poste]→(rep),  
(rep)-[:repond]→(mess),
```



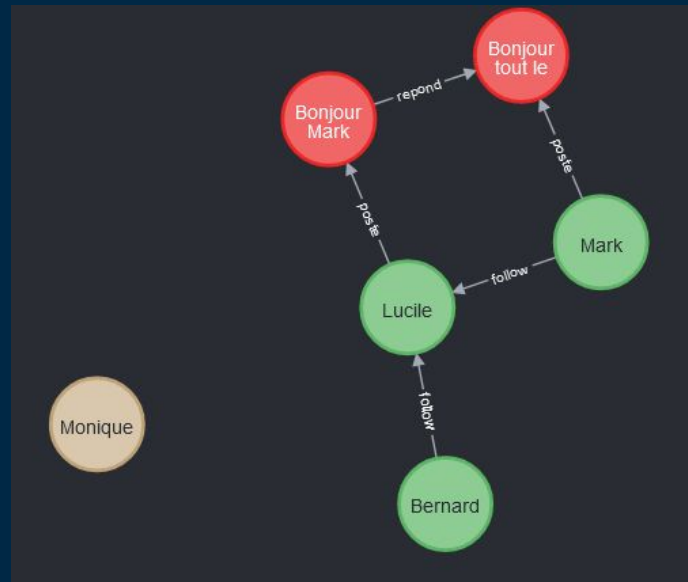
Création des données

3. Démonstration

CRUD (Create, Read, Update, Delete)

```
(monik:Personne {prenom:"Monique", age:34}),  
(:Personne:utilisateur {prenom:"Bernard", age:27})-[:follow]→[l]
```

Création de nouvelles données

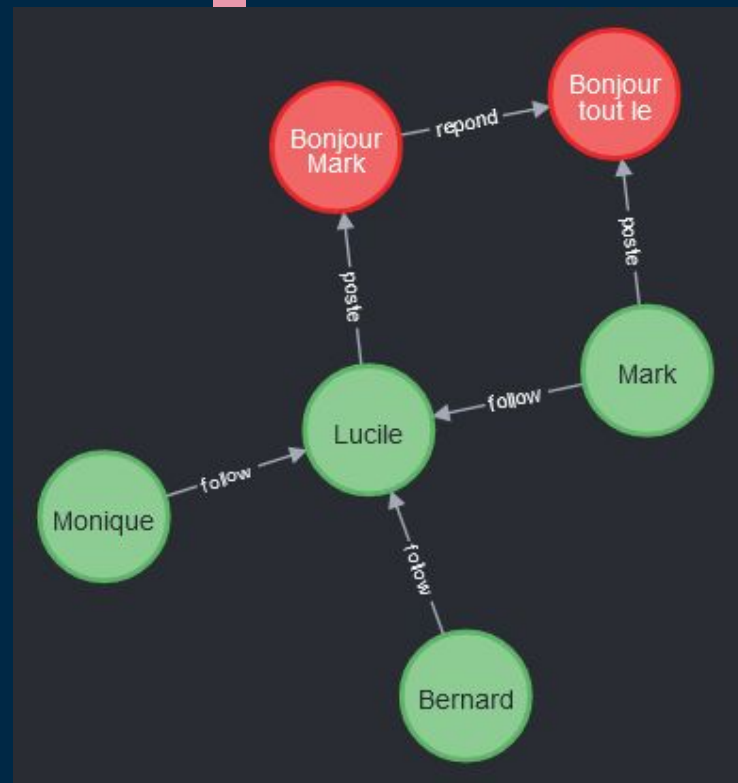


3. Démonstration

CRUD (Create, Read, Update, Delete)

```
MATCH (p:Personne {prenom:'Monique'})  
MATCH (u:utilisateur {prenom:"Lucile"})  
SET p:utilisateur  
CREATE (p)-[:follow]→(u)
```

Modification des données



3. Démonstration

CRUD (Create, Read, Update, Delete)

```
MATCH (n) DETACH DELETE n
```

Suppression de toutes les données

3. Démonstration

Comment accéder avec un programme ?



Conclusion

Neo4j permet :

- Manipulation de grandes données connectées
- Supporter de nombreux modèles complexe et flexible
- Performances exceptionnelles pour la recherche de données :
 - par le chemin le plus court
 - par la détection de boucles
 - autres ...



Conclusion

Neo4j est très rapide, puissant et flexible.

Neo4j is Awesome!





MERCI !

Chen Zhihao, Maxence Hennekein,
Baptiste Petrau, Loïc Dumay

