



Introduction aux bases NoSQL de type graphe

Concepts des bases NoSQL de type graphe
Modélisation
Cas d'utilisation
Correspondance Relationnelle/Graphe

Concepts des bases NoSQL de type graphe

Nœud ou Sommet(Node or Vertex)

- Nom
- **Propriété (Property)** de type clef-valeur
La valeur est de type simple :
 - Numérique
 - Chaîne caractères,
 - Date,
 - Booléen,
 - Liste...

Un Nœud ne contient pas d'autres nœuds

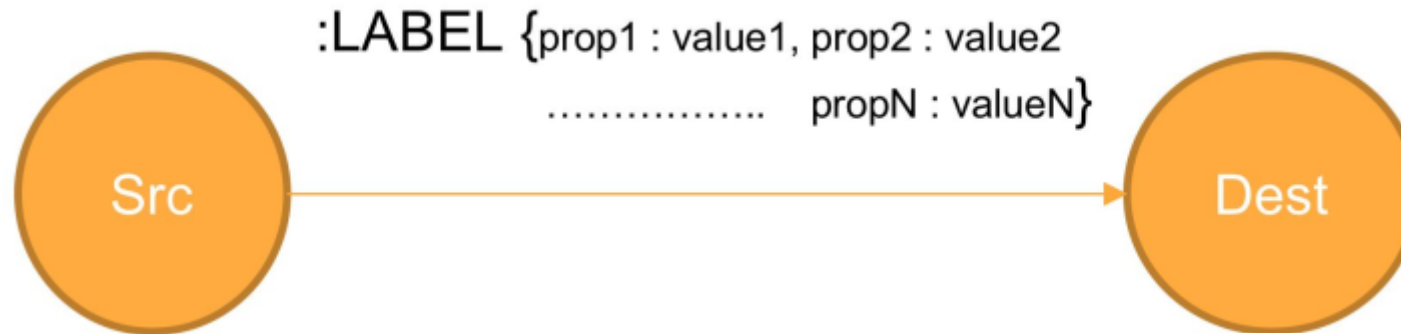
Label: Nom donné à un ensemble de Nœuds



[Telecom ParisTech] Slides Cours Neo4j

Arête ou Relation (Edge)

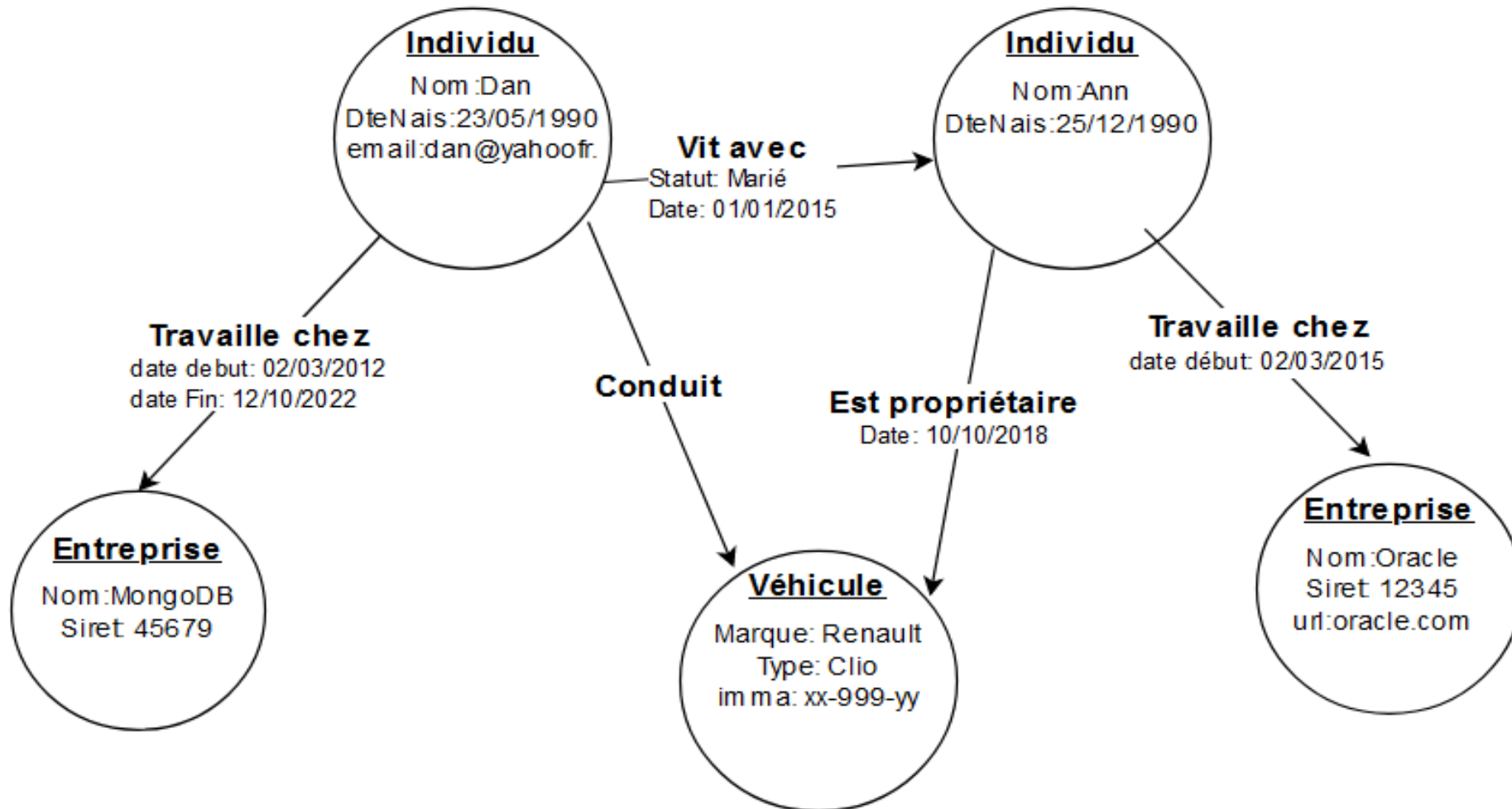
- Relie deux nœuds
- Peut comporter aussi des propriétés.
- Est **orientée** :
comporte
 - un nœud de départ (**start node**)
 - un nœud d'arrivée (**end node**).



[Telecom ParisTech] Slides Cours Neo4j

Label: Nom donné à un ensemble de **Relations**

Concepts des bases NoSQL de type graphe





Concepts des bases NoSQL de type graphe

Un **chemin élémentaire** est un chemin ne passant pas deux fois par un même sommet , c'est-à-dire dont tous les sommets sont distincts.

Un **chemin simple** est un chemin ne passant pas deux fois par un même arc, c'est-à-dire dont tous les arêtes sont distincts.

Un **circuit** est un chemin dont les deux extrémités sont identiques.

La **longueur d'un chemin** est le nombre d'arêtes du chemin,

Parcourir un graphe consiste à visiter ses sommets, en suivant les arêtes qui les relient...

- Parcours en profondeur
- Parcours en largeur

Plus d'infos : <https://info.blaisepascal.fr/nsi-parcours-dun-graphe/>



Modélisation

Un nœud contient des attributs de type simple d'où la nécessité de découper son modèle de données en petites entités.

Un nœud peut être **typé(s)** (un ou plusieurs labels).

Les propriétés n'étant pas forcément normalisées, on peut avoir des champs différents pour un même type de nœud ou de relation.

Dans les faits, il est préférable de normaliser un minimum les entités.

Les bases de données orientées graphe ont une modélisation de données naturelle et intuitive. De ce fait, des personnes fonctionnelles peuvent plus facilement comprendre et aider à cette modélisation.



Graphe: Cas d'utilisation

Modèle particulièrement bien adapté aux situations suivantes :

- Modélisation des données provenant des réseaux sociaux (Twitter, Facebook...)
- Moteur de recommandation (vous êtes sûrement intéressés par tel ou tel objet car beaucoup de vos amis et des amis de vos amis le sont aussi)
- Détection de la fraude organisée (détection de réseaux de fraude)
- Catalogue de produits
- ...

Plus globalement dans le cas de données fortement connectées.

Modèle pas adapté aux:

- Applications avec des forts besoins de mises à jour
- Données non interconnectés



Correspondance Relationnelle/Graphe

Base de données relationnelles	Base NoSQL de type Graphe
Table	Graphe
Ligne	Nœud
Colonne (table)	Propriétés (Nœud et relation)
Contrainte de type FK	Relation
Jointure entre tables	Relation entre nœuds

Correspondance Relationnelle/Graphe

Student

Name	Country	Hair	University
Andrei	1	Black	1
Alexandre	2	Brown	2
Geoffrey	3	Blond	1
Chafik	4	Black	3
Jonathan	3	Black	3

SELECT

s.name, c.name, u.name

FROM

student s

LEFT JOIN country c **ON** c.ID = s.country

LEFT JOIN university u **ON** u.ID = s.university

WHERE

u.name = 'UPMC'

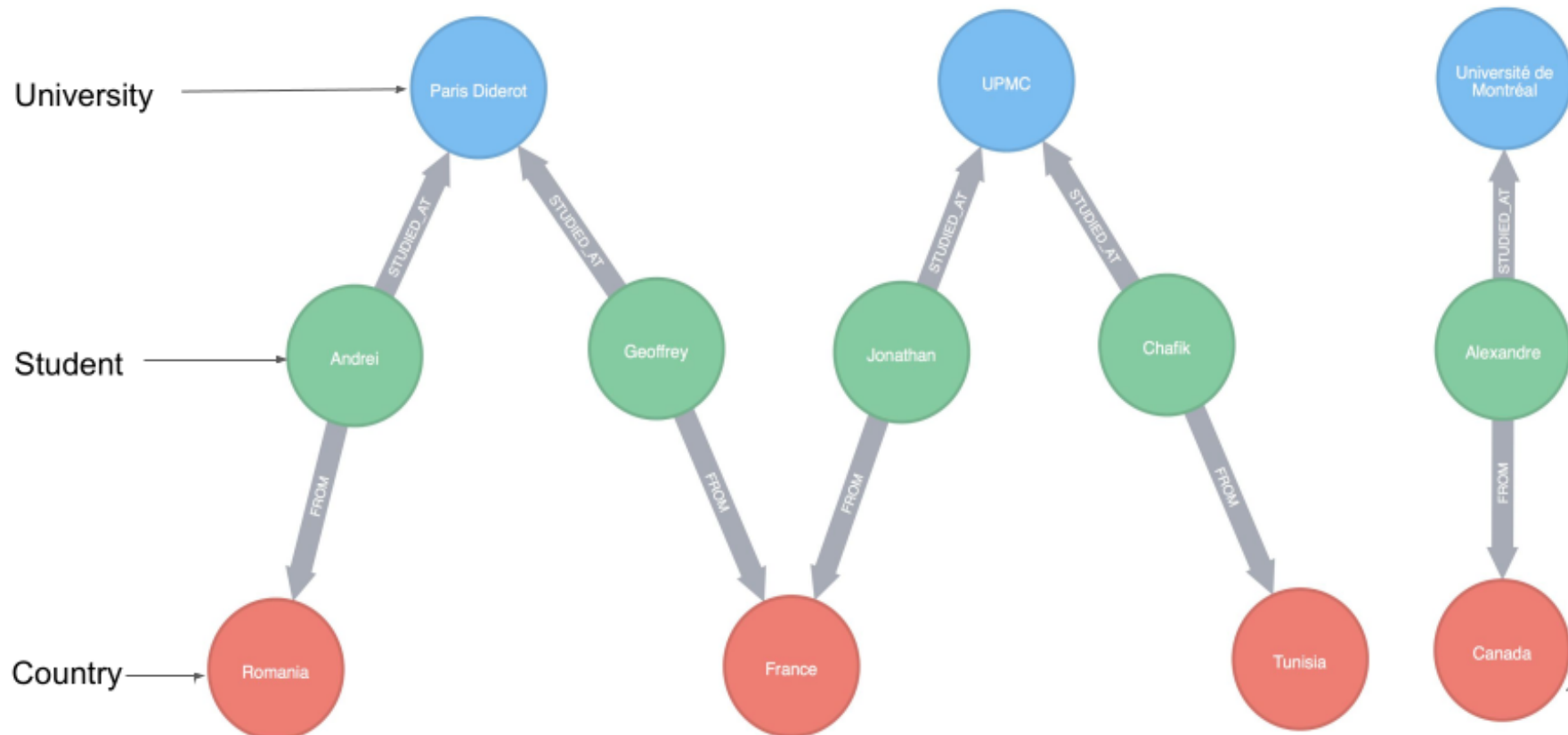
Country

ID	Name	Capital city
1	Romania	Bucharest
2	Canada	Ottawa
3	France	Paris
4	Tunisia	Tunis

University

ID	Name
1	Paris Diderot
2	Université de Montréal
3	UPMC

Correspondance Relationnelle/Graphe



MATCH

`(u:University{name:"UPMC"})<-[:STUDIED_AT]-(s:Student)-[:FROM]->(c:Country)`

RETURN

`s.name AS Student, c.name AS Country, u.name AS University`