

## Chargement d'une base de données externe

Afin de travailler sur une base de données externe vous devez importer ces données grâce à la commande LOAD CSV.

Ex : *LOAD CSV WITH HEADERS FROM*  
*"https://needemand.com/xerf457878/DonateurUK.csv"*

Rien de mieux que la mise en pratique. Donc, nous allons construire une base de données graphique des dons politiques. Les données que nous allons utiliser sont les dons en argent effectués entre Janvier 2010 et Juin 2015 aux partis politiques britanniques.

Nous allons transformer les données brutes CSV dans une base de données graphique qui montre les donateurs et les bénéficiaires en tant que nœuds et la relation entre eux, mais conserve les autres détails tels que le montant du don, dates, etc. comme des propriétés au sein de la base de données.

Petit rappel.

### création d'un nœud :

*CREATE (x:Label {propriété:"Valeur"})*

La lettre au début du code ( "x") est d'attribuer une étiquette de raccourci à mon nœud. Ce n'est pas obligatoirement une lettre, il peut être tout ce que vous voulez!

Le label est d'attribuer un type à votre nœud. Cela peut-être une personne, un objet, un lieu en fonction de votre jeu de données.

La propriété (en minuscules) est ce que j'appelle un attribut que je veux attribuer à ce nœud. Cela pourrait être un nom, un âge, un numéro de référence - dépend de l'ensemble de données que vous utilisez. Vous décidez de son nommage.

La «valeur» est la valeur de ma propriété. . Par exemple, si elle était un nom, je pourrais mettre la valeur en tant que "Driss". Nous verrons cela en action dans un instant.

*create(p:personne{nom:"Driss"})*  
*create(g:gateau{nom:"éclair au chocolat"})*

### création d'une relation :

La relation est écrite entre crochets dans le centre.

Les espaces entre les mots doivent avoir un underscore \_ entre eux.

*create (p)-[r:mange\_tout] → (g)*

*# et ça c'est vrai ! j'aimerais bien*

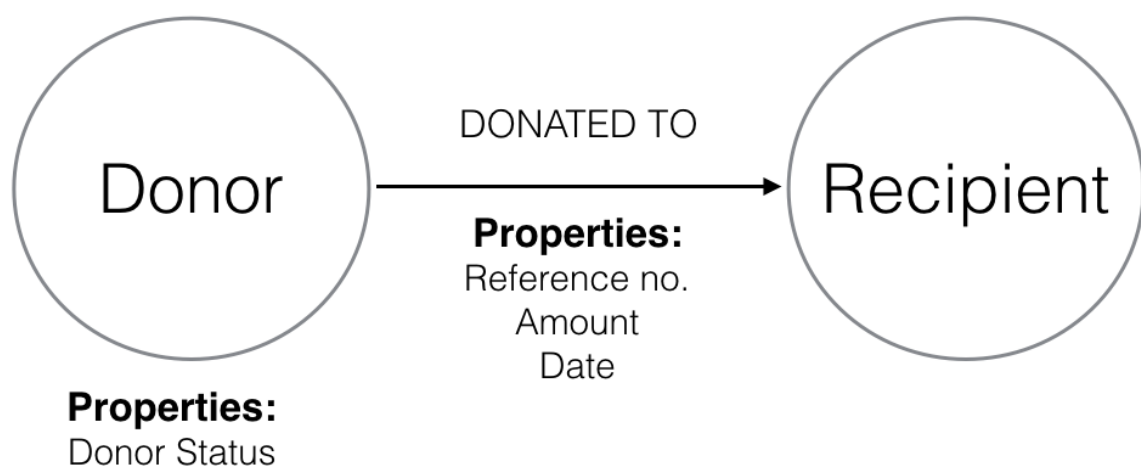
Donc notre nouvelle bd

ECRef	EntityName	Value	DonorName	DonorStatus	ReceivedDate
C0196265	Mr Tom Watson MP	£4,000.00	Ms Joanne Rowling	Individual	11/05/15
C0196253	Conservative Middle East	£15,000.00	Allam House Ltd	Company	22/05/15
C0196254	Conservative Middle East	£15,000.00	International Hospit	Company	26/05/15
C0196251	The Rt Hon Tessa Jowell I	£5,000.00	Mr Trevor Beattie	Individual	07/05/15
C0196255	Labour Women's Network	£10,000.00	Ms Daphne Barbara	Individual	16/04/15
C0196250	The Rt Hon Tessa Jowell I	£5,000.00	Mr Paul Myners	Individual	30/04/15
C0196252	The Rt Hon Tessa Jowell I	£10,000.00	Mr Mervyn Davies	Individual	30/04/15
C0196262	The Rt Hon David Lammy	£5,000.00	Beth Blood	Individual	26/04/15
C0165774	Focus on Scotland	£10,000.00	Patersons of Green	Company	23/04/15
C0194779	Labour People Ltd	£10,000.00	Gina Miller	Individual	17/04/15
C0165777	Movement for Change	£50,000.00	Mr David Sainsbury	Individual	20/04/15
C0165769	Focus on Scotland	£10,000.00	Sir Jack Harvie	Individual	15/04/15
C0165768	Progress Ltd	£65,000.00	Lord David Sainsbur	Individual	30/03/15
C0165770	Focus on Scotland	£10,000.00	Dr Vera Weisfled	Individual	13/04/15
C0165771	Focus on Scotland	£10,000.00	James M McAlpine	Individual	10/04/15
C0196259	The Rt Hon David Lammy	£5,000.00	Mr Killian Hurley	Individual	06/03/15
C0165776	Focus on Scotland	£10,000.00	Rockdale Consultan	Company	07/04/15
C0165772	Focus on Scotland	£15,000.00	Eric N Smith Ltd	Company	07/04/15
C0165773	Focus on Scotland	£10,000.00	Mr Kenneth Harvie	Individual	07/04/15
C0165775	Focus on Scotland	£10,000.00	Ronald McLetchie	Individual	06/04/15
C0194780	Labour People Ltd	£20,000.00	Nick Harrison	Individual	01/04/15
C0161116	Conservative Christian Fe	£75,000.00	Mr Michael Farmer	Individual	01/04/15
C0165541	Liberal Democrats	£2,514.60	Lord na Bradshaw	Individual	31/03/15
C0193852	Conservative Party	£2,200.00	Mr David Nicholls	Individual	31/03/15
C0194367	Conservative Party	£2,500.00	Mr Howard P Shore	Individual	06/03/15
C0194379	Conservative Party	£5,000.00	Paul John Construct	Company	31/03/15
C0194545	Conservative Party	£2,000.00	Mr John Kinder	Individual	30/03/15
C0194196	Conservative Party	£5,000.00	Lord Michael S Farn	Individual	31/03/15
C0194088	Conservative Party	£75,000.00	Mr Michael Tory	Individual	31/03/15
C0194526	Conservative Party	£2,500.00	FPE Global Ltd	Company	02/03/15
C0194475	Conservative Party	£10,000.00	Ms Valerie Metcalfe	Individual	31/03/15
C0193845	Conservative Party	£5,657.48	Mr Daniel L Poulter	Individual	31/03/15

**Needemand**



En gardant à l'esprit que les deux nœuds et les relations peuvent avoir des propriétés, le modèle de données que je veux créer:



**Needemand**

Neo4j va créer aveuglément les noeud pour nous.

Et beaucoup d'autres ensembles de données de la même chose ou le nom peut apparaître plusieurs fois. En d'autres termes, il nous crée des doublons!

L' utilisation de CREATE ne me dira pas ce qu'il y a dans un fichier CSV. Il ne dit pas que la même personne aurait donné plusieurs fois ou à plusieurs entités.

Au lieu de **CREATE** nous allons utiliser **MERGE** .

MERGE passera chaque ligne du CSV et va rechercher les doublons. Si elle constate des instances de la même chose, elle va les fusionner en un seul noeud, comme le nom d'un homme politique par exemple.

Si elle ne trouve pas un doublon, elle va créer un nouveau noeud pour cette « datapoint ».

Nous allons substituer le CREATE pour MERGE et l' ajouter à notre commande load csv.

Parce que nous apportons nos données à partir d' un fichier CSV, nous avons besoin d'ajouter le terme line avant notre valeur ainsi, Neo4j sait à partir d'où commence les données.

NE PAS exécuter pour l' instant.

```
LOAD CSV WITH HEADERS FROM
"https://needemand.com/xerf457878/DonateurUK.csv" AS line
MERGE (r:Recipient {recipientName: line.EntityName})
MERGE (d:Donor {donorName: line.DonorName})
```

## Ajout de propriétés

Jusqu'à présent, nous avons pris soin de charger les données et la création de nos nœuds.

Le fichier CSV contient des données supplémentaires, certaines propriétés, mais que vous voulez être en mesure de récupérer ou utiliser lors de l'interrogation de vos données.

Nous savons déjà comment ajouter des propriétés - {recipientName: line.EntityName} .

Pour ajouter plusieurs propriétés, il suffit de faire la même chose dans les crochets séparés par une virgule comme ceci: {donorName: line.DonorName, donorStatus: line.DonorStatus}

Ne pas oublier les nœuds et les relations peuvent avoir des propriétés. Je vais ajouter trois propriétés à ma relation.

```
CREATE (d)-[:DONATED_TO {ref: line.ECRef, amount: line.Value, date:
line.ReceivedDate}]->(r) .
```

**Needemand**

## Mettre tout ensemble

Mettons maintenant tout cela ensemble et fabriquons notre base de données!

```
LOAD CSV WITH HEADERS FROM
"https://needemand.com/xerf457878/DonateurUK.csv" AS line
MERGE (r:Recipient {recipientName: line.EntityName})
MERGE (d:Donor {donorName: line.DonorName, donorStatus: line.DonorStatus})
CREATE (d)-[:DONATED_TO {ref: line.ECRef, amount: line.Value, date:
line.ReceivedDate}]->(r)
```

Lancez l'exécution (bouton tout à droite en forme de triangle). Cela peut prendre plusieurs minutes en fonction de la vitesse de votre ordinateur car Neo4j passe par chaque ligne et la vérification des doublons avant de créer les nœuds et les relations.

## Requêtes de base en Neo4j

Vous avez peut-être remarqué que lorsque vous cliquez sur une étiquette dans la barre latérale, il ramène au hasard 25 nœuds. Pour effectuer des recherches plus spécifiques, nous avons besoin de savoir comment interroger les données.

Neo4j est extrêmement puissant et peut être utilisé pour les requêtes très complexes.

Ce [refcard Cypher](#) vous aidera à vous familiariser avec les principes de base, mais pour l'instant, voici quelques exemples pour vous aider à démarrer.

Toutes les requêtes commencent par MATCH et se terminent par RETURN (sinon il va exécuter la requête mais pas vous afficher le résultat!).

Pour rechercher certaines conditions, nous utilisons la WHERE commande.

Une requête simple ressemblerait à quelque chose comme ceci:

```
MATCH x WHERE x.recipientName ="Caroline Lucas"
```

En d'autres termes, on recherche quelque chose où le nom du destinataire est Caroline Lucas.

**Needemand**

Expressions régulières :

Voyons modifiant notre requête Cypher pour trouver Caroline Lucas, ajoutant =~ et .\* pour la rendre moins exclusive dans sa recherche:

```
MATCH r WHERE r.recipientName =~ ".*Caroline Lucas.*"
RETURN r
```

Cette fois, elle apparaît! Double-cliquez sur « elle » pour voir ses donateurs.

La requête Cypher dit "Faites correspondre quelque chose où le nom du bénéficiaire est égal ou approximativement à Caroline Lucas".

Le. \* De chaque côté rend la correspondance large, de sorte que nous pourrions assimiler cela à une façon de dire «contient». Vous pouvez l'utiliser pour une partie d'un nom ou

sélectionner des lettres pour le rendre encore plus large.

→ Ensuite, essayez de trouver d'autres députés tels que Tom Watson, David Lammy, Mark Reckless et Tessa Jowell

→ Disons que nous voulons rechercher un type spécifique de donateurs, comme une société. Nous avons nommé la propriété 'donorStatus' lors de la mise en place de notre base de données.

Si vous voulez ajouter plusieurs critères, utilisez AND.

→ Essayez de me trouver un donneur qui a donné à la fois au travailist et aux conservateurs.

→ Ou je pourrais vouloir chercher des donateurs qui sont des sociétés et ont donné plus d'une certaine somme d'argent.

Une bonne façon de pratiquer utilisez le refcard mentionné précédemment.

## PS:

- Essayez de ne pas avoir d' espace dans vos entêtes CSV si possible, Neo4j peut mal les gérer.
- Si votre requête implique des chiffres que vous pourriez avoir besoin d'utiliser, vous avez besoin de dire à Neo4j au stade de chargement que la cellule contient un nombre entier.

Exemple le montant d'un don politique. Lors du chargement d'une valeur, il faut caster toInt(amount: line.Value) .

**Se référer également à la refcard et le site Web pour obtenir des renseignements sur le traitement des dates.**

**Needemand**

Un autre petit exercice pour la mise en jambe !

Données issues du site <http://www.regardscitoyens.org/sunshine/>

Suite au scandale Médiateur, la France a adopté en 2011 une Sunshine Act . Pour la première fois, nous disposons de données sur les cadeaux et les contrats accordés aux professionnels des soins de santé par les compagnies pharmaceutiques. Peut-on utiliser la visualisation de graphes pour comprendre ces liens dangereux?

Oui et vous allez le faire;) si si rachel, de même valentin:)

Regards Citoyens, une ONG française, a parsé diverses sources pour construire la première base de données documentant les relations financières entre les fournisseurs de soins de santé et les fabricants de produits pharmaceutiques.

Cette base de données couvre une période de partir de Janvier 2012 à Juin 2014.

Elle contient 495 951 professionnels de la santé (médecins, dentistes, infirmières, sages-femmes, pharmaciens) et 894 sociétés pharmaceutiques. Les contrats et les cadeaux représentent un total de 244 572 645 €.

Les données originales peuvent être trouvées sur le site de Regards citoyens.

Les données sont stockées dans un grand fichier CSV.

Nous allons utiliser la visualisation de graphes pour comprendre le réseau formé par les relations financières entre les sociétés pharmaceutiques et les professionnels des soins de santé.

Premièrement, nous devons tout importer.

(on a fait le travail pour vous, voilà le script d'importation, attention n'oubliez pas de changer le script selon l'emplacement de votre fichier csv) :

[sunshine.anonymes.csv](#)

```
//-----  
//Importer les docteurs  
//-----
```

```
CREATE CONSTRAINT ON (a:DOCTOR) ASSERT a.ID IS UNIQUE;
```

```
USING PERIODIC COMMIT 2000  
LOAD CSV WITH HEADERS FROM "c:\sunshine.anonymes.csv" AS line  
FIELDTERMINATOR ','  
MERGE (a:DOCTOR {ID: line.BENEF_PS_ID})  
ON CREATE SET a.ORIGIN= line.ORIGIN, a.BENEF_PS_QUALIFICATION=  
line.BENEF_PS_QUALIFICATION, a.BENEF_PS_DEPARTEMENT=  
line.BENEF_PS_DEPARTEMENT, a.totalDECL = toInt(line.DECL_AVANT_MONTANT)  
ON MATCH SET a.totalDECL = toInt(line.DECL_AVANT_MONTANT) + a.totalDECL;
```

**Needemand**



```
//-----  
//Importer les laboratoires  
//-----
```

```
CREATE CONSTRAINT ON (a:PHARMA) ASSERT a.NAME IS UNIQUE;
```

```
USING PERIODIC COMMIT 2000  
LOAD CSV WITH HEADERS FROM "c:\sunshine.anonymes.csv" AS line  
FIELDTERMINATOR ','  
WITH line  
WHERE line.LABO IS NOT NULL  
MERGE (a:PHARMA {NAME: line.LABO})  
ON CREATE SET a.totalDECL = toInt(line.DECL_AVANT_MONTANT)  
ON MATCH SET a.totalDECL = toInt(line.DECL_AVANT_MONTANT) + a.totalDECL;
```

```
//-----  
//Importer les dons individuels des laboratoires aux médecins  
//-----
```

```
USING PERIODIC COMMIT 2000  
LOAD CSV WITH HEADERS FROM "c:\sunshine.anonymes.csv" AS line  
FIELDTERMINATOR ','  
MATCH (b:DOCTOR {ID: line.BENEF_PS_ID})  
MATCH (a:PHARMA {NAME: line.LABO})  
CREATE (a)-[:HAS_DONATED {  
    DECL_AVANT_DATE: line.AMNDT_IND,  
    DECL_AVANT_NATURE: line.DECL_AVANT_NATURE,  
    DECL_AVANT_MONTANT: toInt(line.DECL_AVANT_MONTANT),  
    DECL_CONV_OBJET: line.DECL_CONV_OBJET,  
    SOURCE: line.SOURCE  
}]->(b);
```

```
//-----  
//Faire aggrégation des dons des laboratoires aux médecins  
//-----
```

```
USING PERIODIC COMMIT 2000  
LOAD CSV WITH HEADERS FROM "c:\sunshine.anonymes.csv" AS line  
FIELDTERMINATOR ','  
MATCH (b:DOCTOR {ID: line.BENEF_PS_ID})  
MATCH (a:PHARMA {NAME: line.LABO})  
MERGE (a)-[r:IS_LINKED_TO]->(b)  
ON CREATE SET r.numberDECL = 1, r.totalDECL = toINT(line.DECL_AVANT_MONTANT)  
ON MATCH SET r.numberDECL = r.numberDECL +1, r.totalDECL =  
toINT(line.DECL_AVANT_MONTANT) + r.totalDECL;
```

**Needemand**



A vous, répondez aux questions :

\*Identifier le médecin qui a reçu le plus d'argent des compagnies pharmaceutiques.

\*Quelles sont les compagnies pharmaceutiques les plus généreuses.

\*Donnez le montant total des cadeaux donnés par les laboratoires, d'abord la valeur déclarée par les laboratoires puis par les praticiens de santé et enfin la somme dans la réalité

labo : 201 944 695

praticien : 114 632 930

réel : 246 650 564

4) Donnez le nombre de contrat établi entre les laboratoires et les praticiens ainsi que le nombre de cadeaux offerts :

5) Quels sont les médecins qui acceptent des cadeaux de différents laboratoires  
( !!requête longue à s'exécuter, le code suffit)

6) Quels médecins fraudent en ne déclarant rien, indiquez bien dans les résultats  
« FRAUDE » comme valeur si ils ne déclarent rien.

7) Donnez la moyenne, le max et le minimum des sommes déclarées par les médecins

Pourquoi la moyenne est différente de la valeur du quartile médian:)

**Needemand**