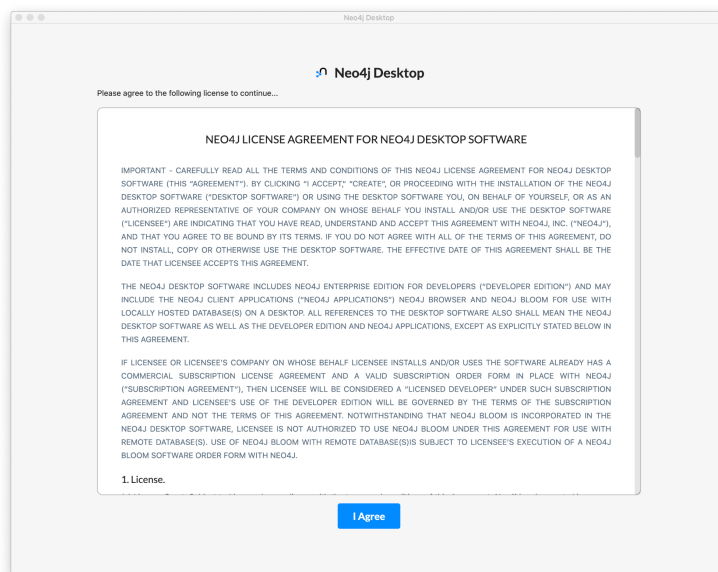


# Neo4j

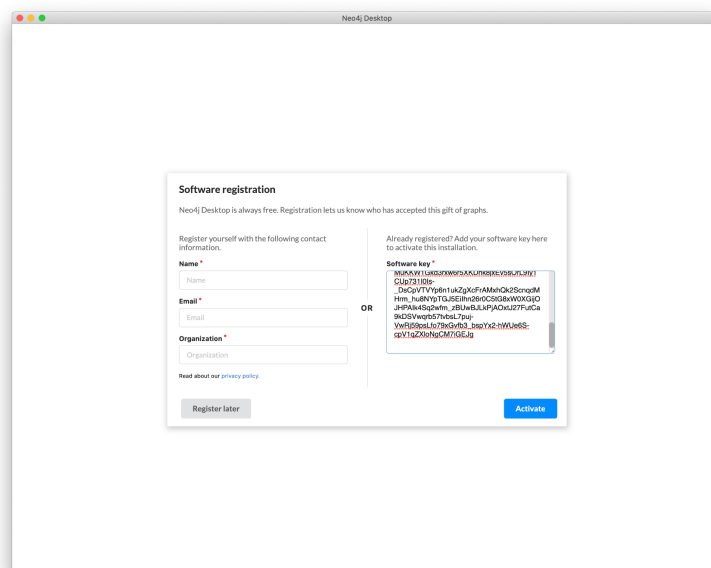
## 1. Installation de Neo4j Desktop

Pour installer Neo4j, rien de plus simple que de récupérer la version Desktop sur [Neo4j.com](https://neo4j.com) puis de suivre les procédures affichées.

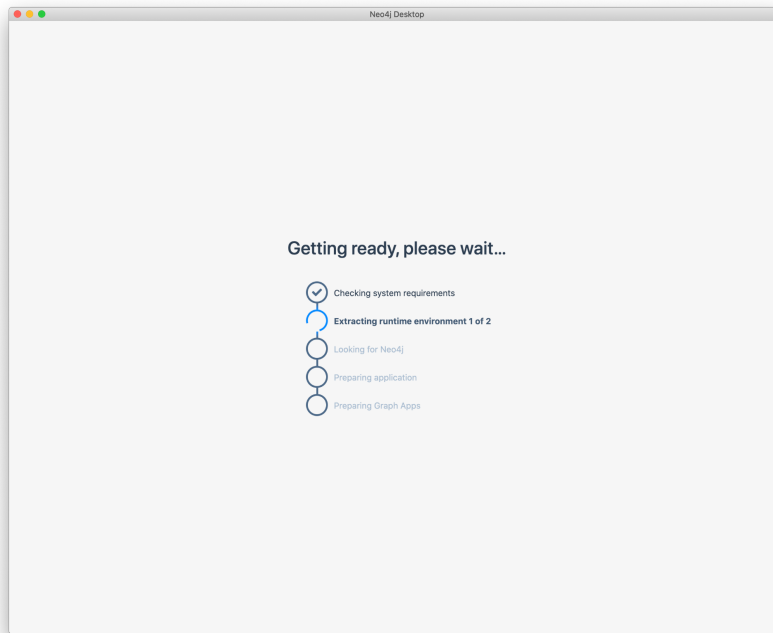
Le lien de téléchargement est : <https://neo4j.com/download-neo4j-now/>



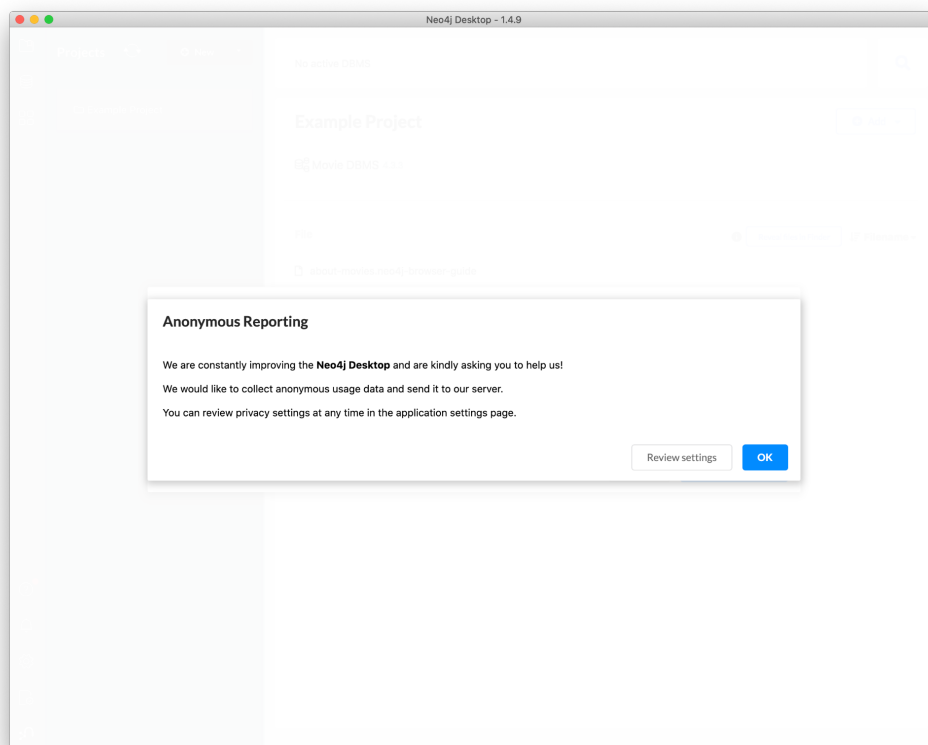
*Capture de la partie de validation de la licence d'exploitation de Neo4j Desktop.*



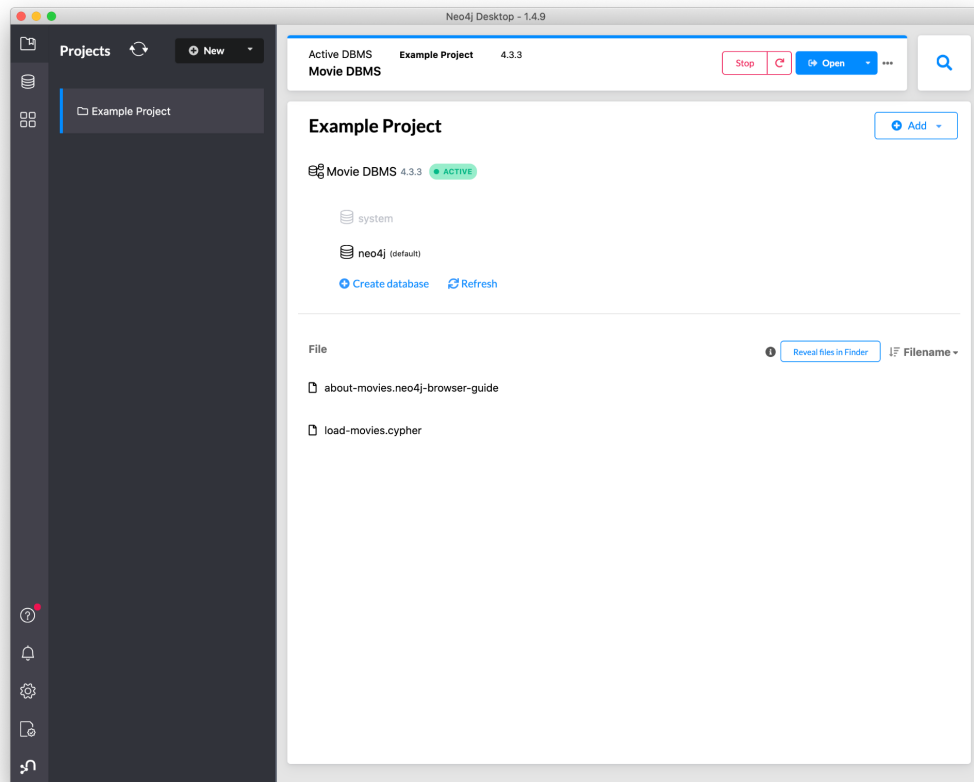
*Capture de la partie de validation de la clé d'utilisation de Neo4j Desktop fournit à l'inscription sur neo4j.com.*



*Capture de la partie de d'installation automatique de Neo4j Desktop.*



*Détection d'une erreur durant la dernière phase d'installation.  
Celle-ci est ignorable et ne dérange pas une utilisation rapide de Neo4j.*



*Capture de la page principale de Neo4j Desktop.*

## 2. Utilisation de Neo4j

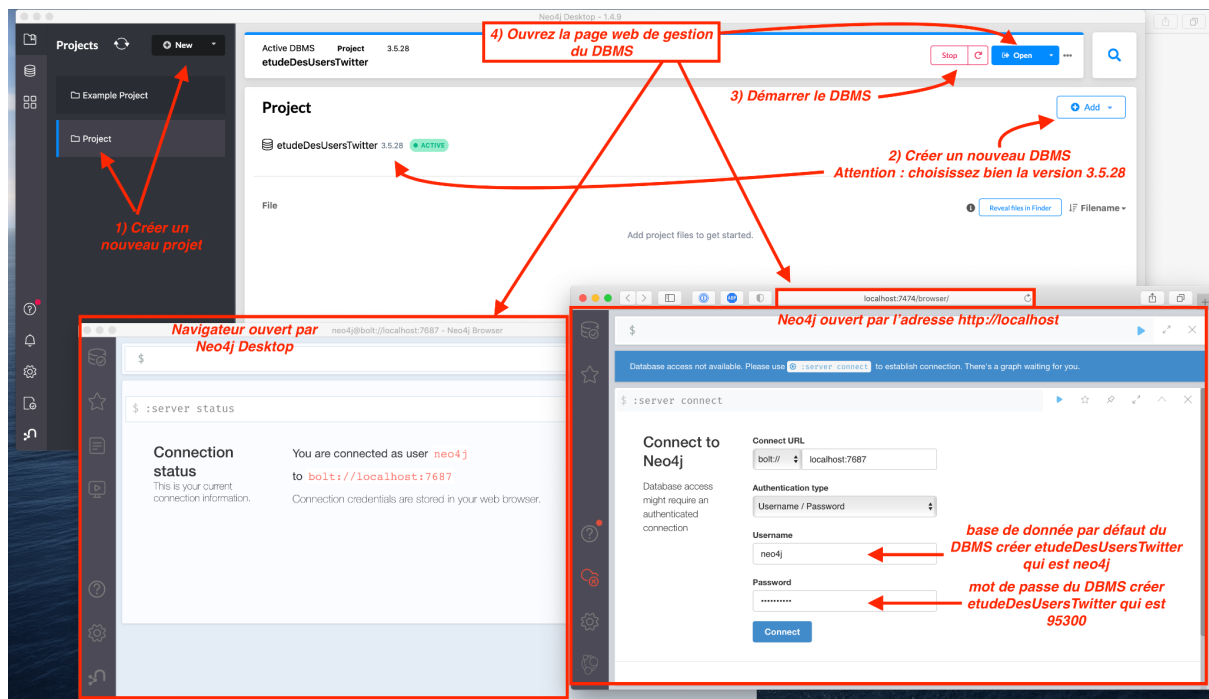
### A. Création d'un nouveau Projet et DBMS

Après avoir installé Neo4j, nous allons créer un nouveau projet.

Dans ce projet il faut créer un nouveau **DBMS** (Data Base Management System).

Pour ce faire, choisissez comme nom **etudeDesUsersTwitter**, en mot de passe **esiee95300** et en version de DBMS la **3.5.28**. Puis vous pouvez le démarrer.

Une fois le nouveau DBMS créé, nous allons avoir besoin d'une nouvelle base de données, afin de la différencier de celle par défaut. Nous allons la nommer **users**.



## B. Les interactions de base

### A) La création de quelques données.

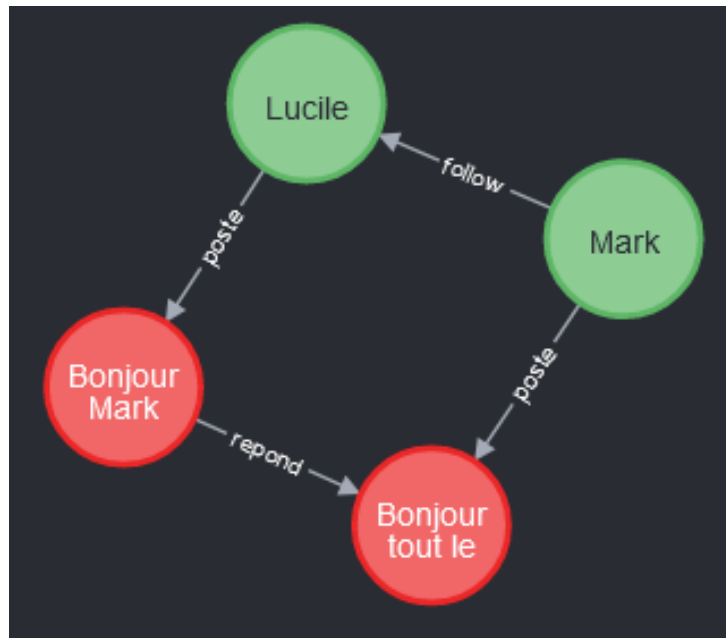
```
CREATE
(m:Personne:utilisateur {prenom:"Mark", age:42}),
(mess:Tweet {message:"Bonjour tout le monde", like:1}),
(l:Personne:utilisateur {prenom:"Lucile", age:21}),
(rep:Tweet {message:"Bonjour Mark", like:21}),
(m)-[:poste]→(mess),
(m)-[:follow]→(l),
(l)-[:poste]→(rep),
(rep)-[:repond]→(mess),
```

*Sur cette capture vous pouvez distinguer la création de nouveau nœuds et relations.*

### B) La lecture des données

```
MATCH (n) RETURN n
```

*Cette commande ci-dessus permet d'afficher tous le contenu de la base de données.*



*Premier résultat graphique de la base de données.*

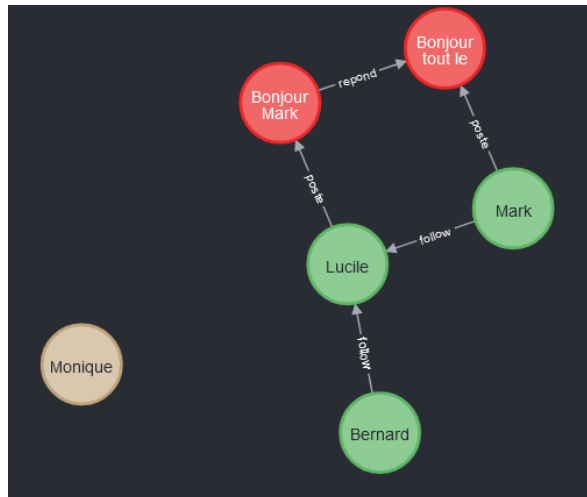
### C) La mise à jour de données

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

Le but de cette capture est de créer deux nouvelles personnes dont une première sans aucun lien et la seconde avec un lien.

**MATCH (n) RETURN n**

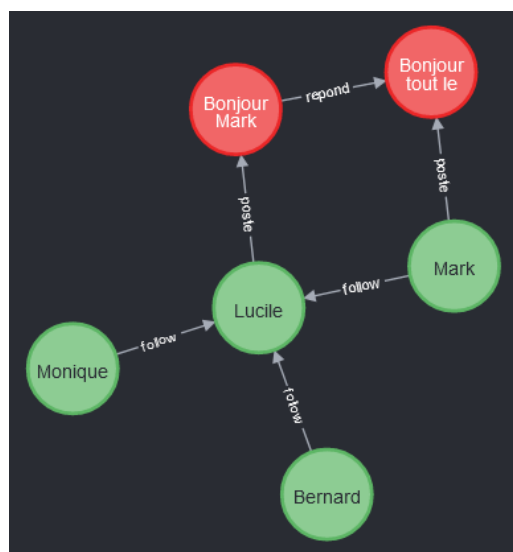
On affiche le résultat complet.



Résultat de la commande d'affichage complet.

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

Maintenant, on veut relier toutes les personnes.



Résultat de la commande par affichage complet.

#### D) La suppressions de données

Pour supprimer des données il est important de les détachés avant.

Donc par affichage, on détache les données que l'on cherche, puis on les supprime.

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

*Cette commande permet de supprimer la totalité des données.*

### 3. Pour aller plus loin

Nous mettons à disposition un github regroupant les essais de programme communiquant avec une base de données Neo4j.

Celui-ci se trouve ici : [https://github.com/ldumay/esiee\\_2021\\_bdd\\_nosql\\_neo4j](https://github.com/ldumay/esiee_2021_bdd_nosql_neo4j)

Dans le cas où le git ne serait plus accessible, le projet est sauvegardé par Loïc DUMAY.

- Via Github : <https://github.com/ldumay/>
- Via Web : <https://ldumay.fr/>