NoSQL - DIA2

MANNAI Hasna (≈ 33%)
CHENIK Yassine (≈ 33%)
BOUCHIBA Emine (≈ 33%)

RAPPORT
TP Neo4J

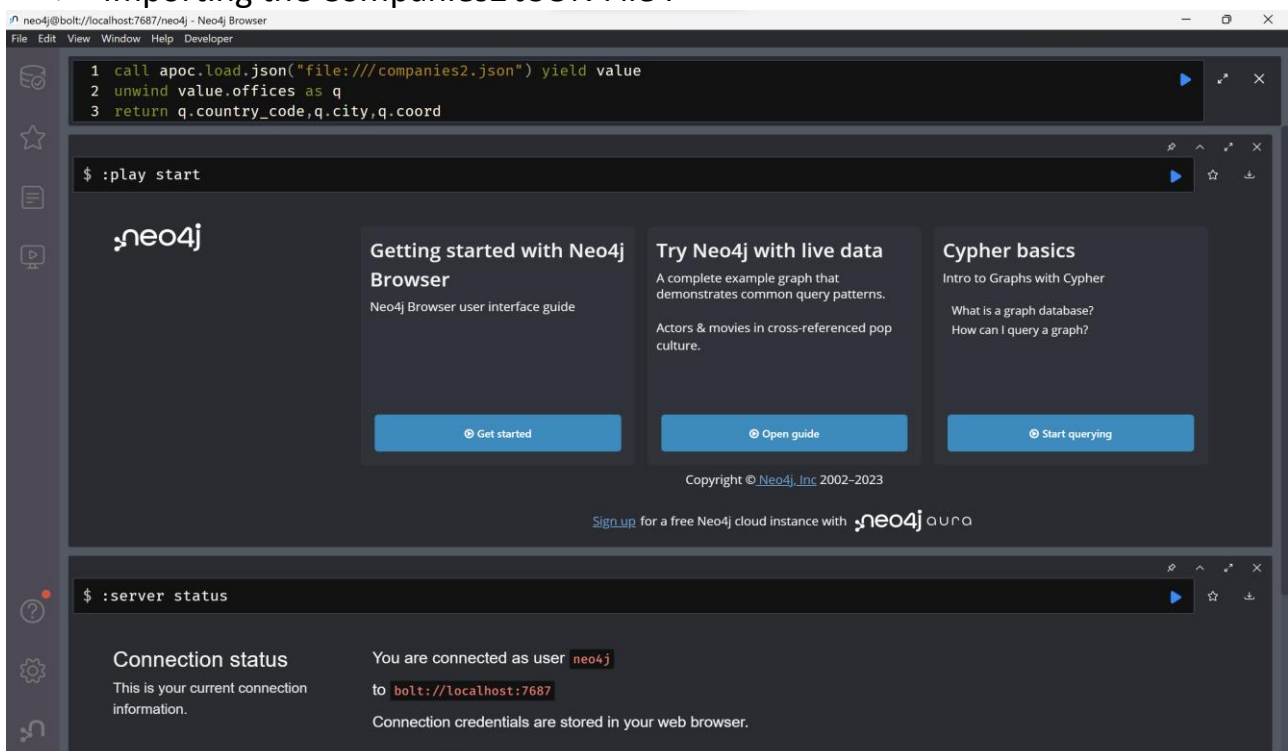# TABLE DES MATIERES

Importation du ficher JSON en utilisant APOC

Neo4J :

➢ Installation of APOC :
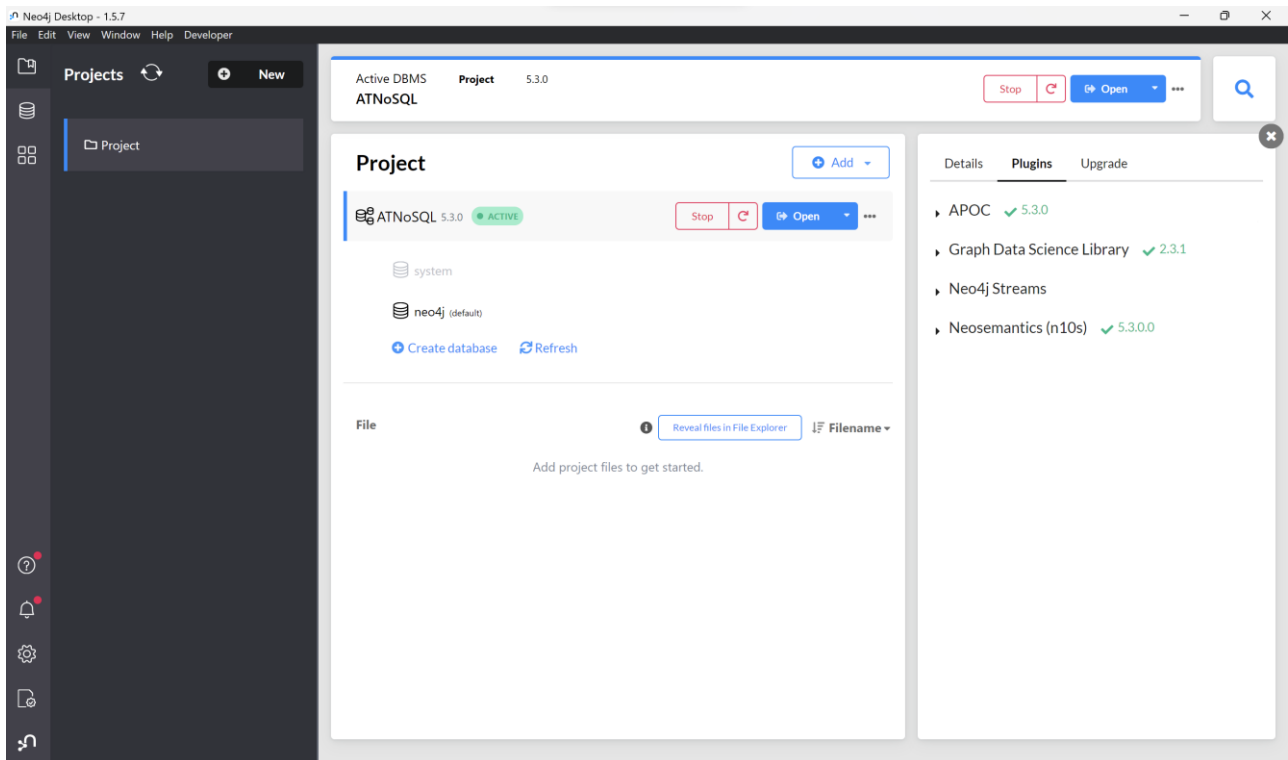
➢ Importing the Companies2 JSON File :

Simple Queries

1. Get all the companies that have a category_code=''nanotech'' :

✓ Code :

```
MATCH (c:companies)
WHERE c.category_code = 'nanotech'
RETURN c
```

✓ *Ici, on veut afficher les entreprises qui sont dans le domaine de la nanotechnologie en affichant le **"name"** et le **"category_code"**.*

2. Get all the Companies that founded in 2008 :

✓ Code :

```
MATCH (c:companies)
WHERE c.founded_year >= 2008 AND c.founded_year <= 2008
RETURN c.name, c.founded_year, c.founded_month, c.founded_day
```

✓ *Ici, on veut afficher les entreprises qui ont été fondés en 2008. en affichant le **"name"**, le **"founded_year"**, le **"founded_month"** et le **"founded_day"**.*

3. Get the number of Companies whith more than 100 employees :

✓ Code :

```
MATCH (c:companies)
WHERE c.number_of_employees >= 100
RETURN c.name, c.number_of_employees
```

✓ *Ici, on veut afficher les entreprises qui ont plus de 100 employés en affichant le **"name"** et le **"number_of_employees"**.*

Simple Queries

4. Get the total amount raised per companies where category_code= "security" :

✓ Code :

```
MATCH (c:companies {category_code: 'security'})
WITH c, c.name as name
LIMIT 1
OPTIONAL MATCH (c)-[:RAISED]->(round)
RETURN name, sum(round.raised_amount) as total_raised
```

✓ *Ici nous cherchons à afficher le **montant total amassé** par les entreprises qui travaillent dans le **domaine de la sécurité**.*

5. Get total money raised for each domain :

✓ **Code :**

```
MATCH (c:companies)
WITH c.category_code as category_code, sum([round in c.funding_rounds |
round.raised_amount]) as total_raised
RETURN category_code, total_raised
ORDER BY total_raised DESC
LIMIT 1
```

✓ *Ici, nous cherchons à afficher le **montant total amassé** par les entreprises du **même domaine**.*

6. Get all companies that starts with an "F" :

➢ Code :

```
MATCH (c:companies)
WHERE c.name STARTS WITH 'F'
RETURN c.name
```

✓ *Ici, on cherche à afficher toutes les entreprises qui commencent par la lettre F. Pour ce faire, on utilise l'option **STARTS WITH** qui nous permet de reconnaître un motif dans une chaîne de caractères et on lui donne comme argument **'F'**. De et on affiche les noms des entreprises correspondantes.*

Complex Queries

1. **Get the number of employees of each Company that have as category_code =''software'' :**

   ✓ Code :

```
MATCH (c:companies {category_code: 'software'})
WITH c.name as name, c.number_of_employees as number_of_employees,
c.category_code as category_code
ORDER BY number_of_employees DESC
LIMIT 10
OPTIONAL MATCH (c)-[:LOCATED_IN]->(city)
WITH name, number_of_employees, category_code, city.name as city_name
RETURN name, number_of_employees, category_code, city_name
```

   ✓ *Ici, on cherche à avoir le **nombre d'employés** travaillant au sein d'entreprises dans le domaine du **Software** et on affiche **le nom de l'entreprise, sa category_code** et le **nombre d'employés**.*

2. **Get the number of products grouped by companies name :**

   ➢ Code :

```
MATCH (c:companies)
WITH c.name as name, c.products as products
UNWIND products as product
WITH name, COUNT(DISTINCT product.name) as total_products
ORDER BY total_products DESC
LIMIT 1
RETURN name, total_products
```

   ✓ *Ici, on veut afficher le nombre de produits provenant de chacune des entreprises.*

Hard Queries

1. Get the total amount of money raised by companies founded in each year :

➢ Code :

```
MATCH (c:companies)
WITH c.founded_year as founded_year, c.funding_rounds as funding_rounds
UNWIND funding_rounds as round
WITH founded_year, SUM(round.raised_amount) as total_raised
ORDER BY founded_year ASC
LIMIT 1
RETURN founded_year, total_raised
```

✓ *Ici, on cherche à obtenir le montant total d'argent collecté par les entreprises fondées chaque année, trié par ordre décroissant du montant total collecté.*