

**MASTERE SPECIALISE  
EXPERT BIG DATA ENGINEER (MS EBEDE)**

**Rapport Projet neo4j**

ADRESSE ET LOCATION STATIONS  
VELOS DE EN FRANCE

*Année 2022/2023*

**Réalisateurs :**

- DAGRA N'GUESSAN ROBERT
- DEDE EDWIGE

# Sommaire

INTRODUCTION :	3
GÉNÉRALITE :	4
Quelques Requêtes et captures:	4
1- Pour créer nœud vélo :	5
2- Pour créer nœud emplacement de location ( Station ):	6
3- Pour créer nœud client :	7
4- Pour créer une relation entre un vélo et un emplacement pour indiquer que le vélo se trouve actuellement à cet emplacement:	8
5- Pour créer une relation entre un client et un vélo pour indiquer que le client a loué le vélo :.....	8
6- Toutes relations :	9
Application:	10
Interface Admin visualisation :	10
Interface Client Recherche station disponible dans une ville :	11
CONCLUSION :	13

## INTRODUCTION :

En France nous assistons à une grande location de vélos dans des stations différentes :

Le but de notre projet sera de :

- Mettre en place une base donnée sur **neo4j** pour collecter les différents flux ou entités
- Mettre en place une plateforme au service de l'admin station et du client pour la visualisation et la recherche de stations de vélo dans les villes de France

Ce rapport contiendra les éléments utilisés pour la réalisation de notre projet.

## GÉNÉRALITE :

Pour une première étape, nous avons pris connaissance du projet, Puis ressortir les nœud et relations nécessaires, dont :

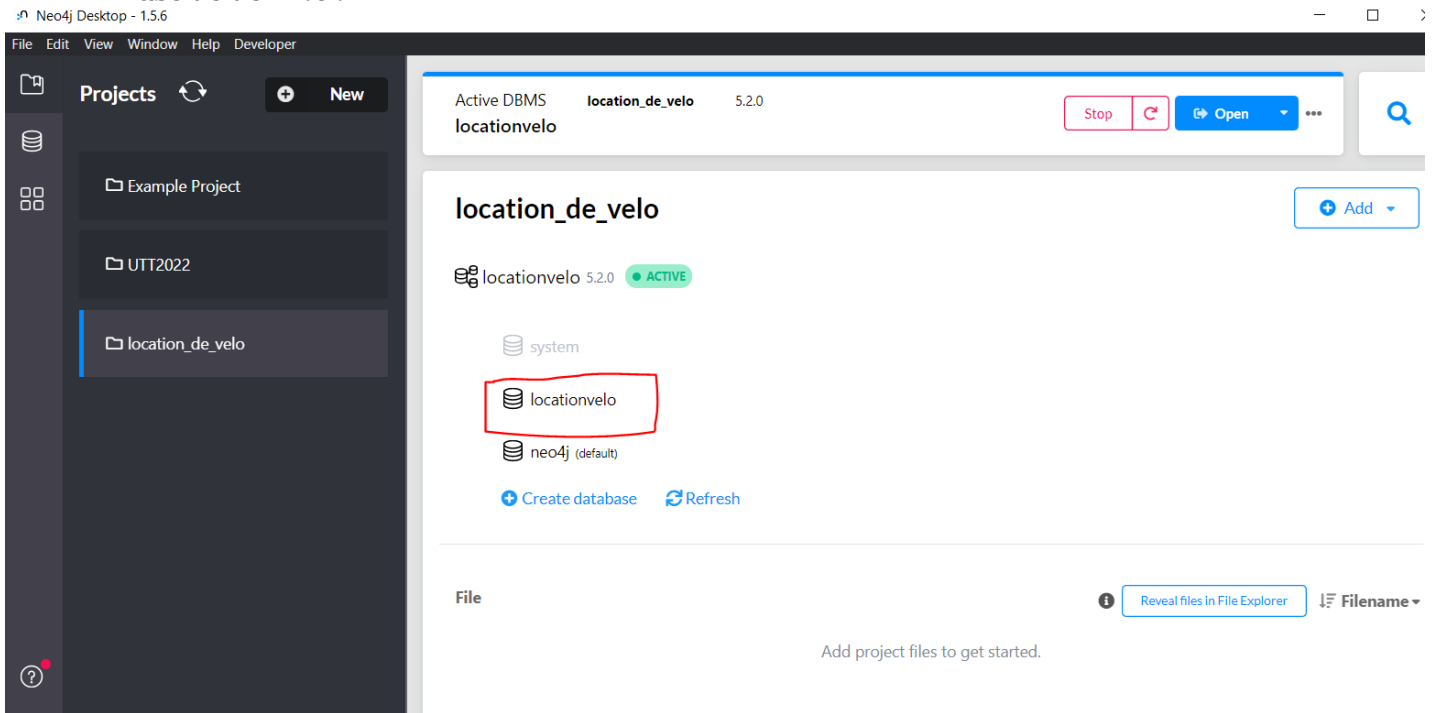
- **Client,**
- **Station**
- **Velo**
- **LOCATION**
- **LOUE**

Ensuite nous avons utilisé les outils, librairies , langage suivants :

- **Neo4j Desktop,**
- **Python**
- **Streamlit**
- **Navigateur WEB (Chrome)**

## Quelques Requêtess et captures:

### Base de donnée :



## 1- Pour créer nœud vélo :

```
// 1- Pour créer nœud vélo :  
  
    CREATE (v:Velo {id: 1, name: "Vélo de montagne", model: "Roadster", size:  
"M"});  
    CREATE (v:Velo {id: 2, name: "Vélo de route", model: "Hamaya", size:  
"S"});  
    CREATE (v:Velo {id: 3, name: "Vélo électrique", model: "Hamaya", size:  
"M"});  
    CREATE (v:Velo {id: 4, name: "Vélo de montagne", model: "Hamaya", size:  
"M"});  
    CREATE (v:Velo {id: 5, name: "Vélo de montagne", model: "Roadster", size:  
"S"});  
    CREATE (v:Velo {id: 6, name: "Vélo électrique", model: "Roadster", size:  
"S"});  
    CREATE (v:Velo {id: 7, name: "Vélo de montagne", model: "Roadster", size:  
"M"});  
    CREATE (v:Velo {id: 8, name: "Vélo de montagne", model: "Hamaya", size:  
"S"});  
    CREATE (v:Velo {id: 9, name: "Vélo électrique", model: "Hamaya", size:  
"M"});  
    CREATE (v:Velo {id: 10, name: "Vélo de montagne", model: "Roadster", size:  
"S"});  
    CREATE (v:Velo {id: 11, name: "Vélo électrique", model: "Roadster", size:  
"M"});  
    CREATE (v:Velo {id: 12, name: "Vélo de montagne", model: "Hamaya", size:  
"S"});  
    CREATE (v:Velo {id: 13, name: "Vélo électrique", model: "Hamaya", size:  
"M"});  
    CREATE (v:Velo {id: 14, name: "Vélo de montagne", model: "Roadster", size:  
"M"});  
    CREATE (v:Velo {id: 15, name: "Vélo de route", model: "Roadster", size:  
"M"});  
    CREATE (v:Velo {id: 16, name: "Vélo de route", model: "Roadster", size:  
"S"});  
    CREATE (v:Velo {id: 17, name: "Vélo de montagne", model: "Hamaya", size:  
"S"});  
    CREATE (v:Velo {id: 18, name: "Vélo de montagne", model: "Roadster", size:  
"M"});  
    CREATE (v:Velo {id: 19, name: "Vélo électrique", model: "Roadster", size:  
"M"});  
    CREATE (v:Velo {id: 20, name: "Vélo électrique", model: "Hamaya", size:  
"M"});
```

Database Information

Use database

locationvelo

Node labels

{41}

Client

Location

Station

Velo

Relationship types

{25}

LOCATION\_A

LOUE

Property keys

address

email

end\_date

id

model

name

size

start\_date

ville

DBMS

Cluster role: primary

locationvelo\$

locationvelo\$ MATCH (n:Velo) RETURN n LIMIT 25

Graph

Table

Text

Code

Vélo de montagne

Vélo de route

Vélo électrique

Vélo de montagne

Vélo de montagne

Node properties

Velo

<id> 18

id 19

model Roadster

name Vélo électrique

size M

## 2- Pour créer nœud emplacement de location ( Station ):

2- Pour créer nœud emplacement de location :

```
CREATE (s:Station {id: 1, name: "Eiffel Tower Station", address: "Champ de Mars, 5 Avenue Anatole France, 75007 Paris" , ville: "Paris"});
CREATE (s:Station {id: 2, name: "Fmille Dagra Station", address: "133 boulevard de la Croix Rousse, 69004 Lyon" , ville: "Lyon"});
CREATE (s:Station {id: 3, name: "UTT MS EBDE Station", address: "Place Alexandre-Israël, 10000 TROYES" , ville: "Troyes"});
CREATE (s:Station {id: 4, name: "Timaelle Maelle Keke Station", address: "31 Rue Gambetta, 78200 Mantes la jolie" , ville: "Mantes la jolie"});
```

**Database Information**

Use database: locationvelo

**Node labels**

(41) Client Location Station Velo

**Relationship types**

(25) LOCATION\_A LOUE

**Property keys**

address email end\_date id model name size start\_date ville

locationvelo\$

locationvelo\$ MATCH (n:Station) RETURN n LIMIT 25

Graph Table Text Code

Timaele Maelle Keke Station

UTT MS EBDE Station

Eiffel Tower Station

Fmille Dagra Station

Illl Station

**Node properties**

Station

<id> 24

address Champ de Mars, 5 Avenue Anatole France, 75007 Paris

id 1

name Eiffel Tower Station

ville Paris

### 3- Pour créer nœud client :

```
// 3- Pour créer nœud client :

CREATE (c:Client {id: 1, name: "Dagra Robert", email: "dagra@gmail.com"});
CREATE (c:Client {id: 2, name: "John Doe", email: "john@example.com"});
CREATE (c:Client {id: 3, name: "Dark Mir", email: "dark@example.com"});
CREATE (c:Client {id: 4, name: "Dagra Sonia", email: "dagra@example.com"});
CREATE (c:Client {id: 5, name: "Christelle Gboblahi", email: "christelle@example.com"});
CREATE (c:Client {id: 6, name: "Rick N'Dri", email: "rick@example.com"});
CREATE (c:Client {id: 7, name: "Elysee N'goran", email: "elysee@example.com"});
CREATE (c:Client {id: 8, name: "Koua Yabi", email: "koua@example.com"});
CREATE (c:Client {id: 9, name: "N'da Victoire", email: "victoire@example.com"});
CREATE (c:Client {id: 10, name: "Mario Jad", email: "mario@example.com"});
CREATE (c:Client {id: 11, name: "Amea David", email: "amea@example.com"});
CREATE (c:Client {id: 12, name: "Hilair Kouassi", email: "hilair@example.com"});
```

The screenshot shows the Neo4j Desktop interface. On the left, the 'Database Information' panel is visible, showing the database 'locationvelo' and node labels 'Client', 'Location', 'Station', and 'Velo'. The main window displays a graph visualization with nodes 'Amea David', 'Dark Mir', 'John Doe', 'Hilair Kouassi', and 'Dagra Robert'. A query window at the top right shows the command: `locationvelo$ MATCH (n:Client) RETURN n LIMIT 25`. The 'Node properties' panel on the right shows details for a 'Client' node: `<id> 28`, `email dagra@gmail.com`, `id 1`, and `name Dagra Robert`.

4- Pour créer une relation entre un vélo et un emplacement pour indiquer que le vélo se trouve actuellement à cet emplacement:

```
MATCH (v:Velo {id: 1}), (s:Station {id: 1})
CREATE (v)-[:LOCATION_A]->(s)
```

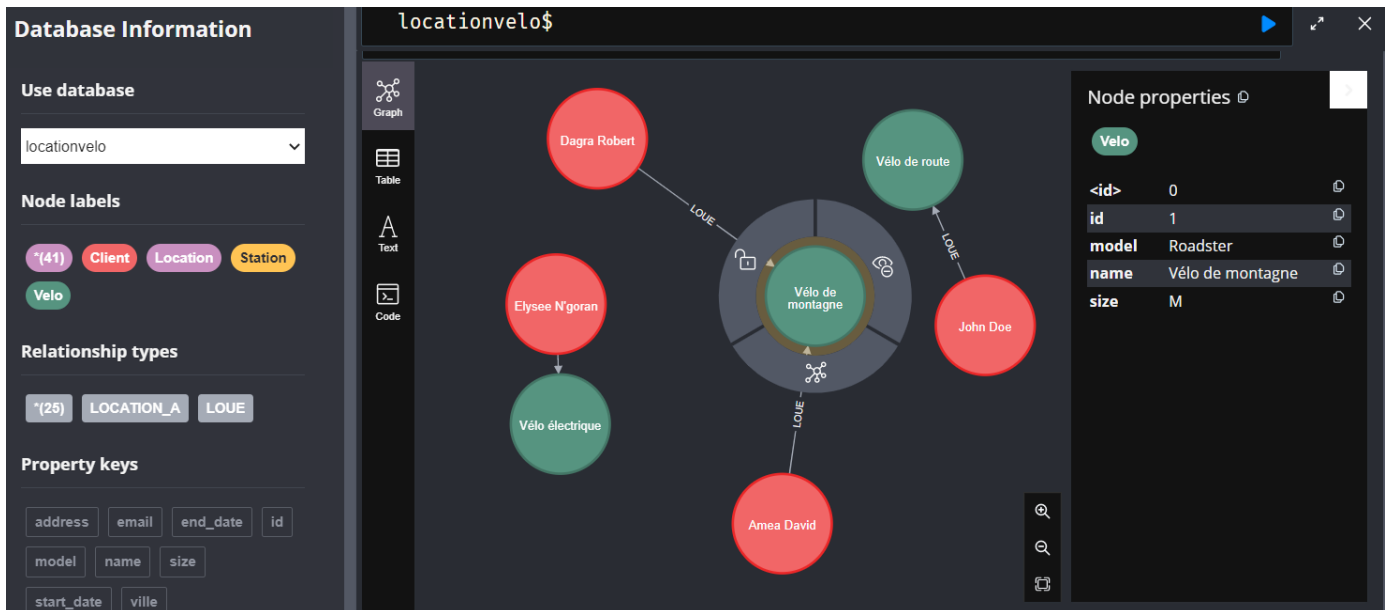
The screenshot shows the Neo4j Desktop interface. On the left, the 'Database Information' panel is visible, showing the database 'locationvelo' and node labels 'Client', 'Location', 'Station', and 'Velo'. The main window displays a graph visualization with a central node 'UTT MS EBDE Station' and several bicycle nodes: 'Vélo de montagne', 'Vélo électrique', 'Vélo de route', and 'Vélo de montagne'. The 'Node properties' panel on the right shows details for a 'Station' node: `<id> 26`, `address Place Alexandre-Israël, 10000 TROYES`, `id 3`, `name UTT MS EBDE Station`, and `ville Troyes`.

5- Pour créer une relation entre un client et un vélo pour indiquer que le client a loué le vélo :

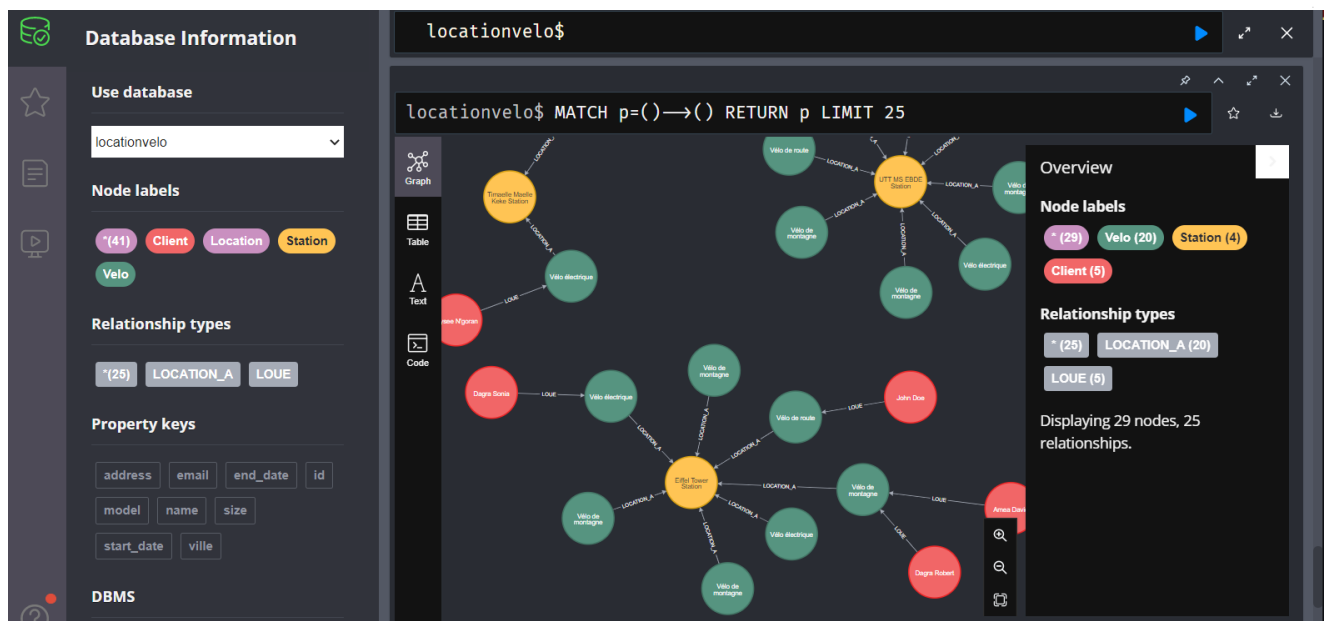
```
MATCH (c:Client {id: 1}), (v:Velo {id: 1})
```



```
CREATE (c)-[:LOUE {start_date: timestamp(), end_date: timestamp()}]->(v)
```



## 6- Toutes relations :



## Application:

**NB :** Pour démarrer l'application lancez l'invite de commande , placez-vous à la racine des fichiers et exécutez

**streamlit run adminspace.py**

**streamlit run clientespace.py**

*Interface Admin visualisation :*

localhost:8501

# ESPACE ADMINISTRATEUR

## VISUALISATION ETAT

### LES DIFFERENTES STATIONS DE VELO

	0	1	2
0	STATION	ADRESSE	VILLES
1	Eiffel Tower Station	Champ de Mars, 5 Avenue Anatole France, 75007 Paris	Paris
2	Fmille Dagra Station	133 boulevard de la Croix Rousse, 69004 Lyon	Lyon
3	UTT MS EBDE Station	Place Alexandre-Israël, 10000 TROYES	Troyes
4	Timaelle Maelle Keke Station	31 Rue Gambetta, 78200 Mantes la jolie	Mantes la jolie
5	Lill Station	23 Rue MORO, 88200 Lill	Lill

### LES CLIENTS

	0	1
0	CLIENT	EMAIL
1	Dagra Robert	dagra@gmail.com
2	John Doe	john@example.com

*Interface Client Recherche station disponible dans une ville :*

# ESPACE CLIENT

## Recherchez une station dans une ville

Entrez la ville: Ex: Paris, Lyon (Respectez la class, s'affiche les informations disponible)

Entrez une ville dans la base s'il vous plait. Ex: Paris, Lyon

# ESPACE CLIENT

## Recherchez une station dans une ville

Entrez la ville: Ex: Paris, Lyon (Respectez la class, s'affiche les informations disponible)

Paris

	0	1	2
0	STATION	ADRESSE	VILLE
1	Eiffel Tower Station	Champ de Mars, 5 Avenue Anatole France, 75007 Paris	Paris

## CONCLUSION :

Ce projet avait donc pour but principal de de faire une base de donnée et la conception une application web pour interagir.

Résultat constaté dans les fichiers et dossiers :

- Exportation de ma base de données en JSON et CSV voir : **DAGRA\_et\_EDWIGE/neo4 /bd/ locationvelo.json** et **locationvelo.csv**
- Application streamlit voir **DAGRA\_et\_EDWIGE/neo4j/app/adminespace.py** et **neo4j/app/clientespace.py**
- Requête cyther voir **DAGRA\_et\_EDWIGE/neo4j/script.txt**
- Rapport voir **DAGRA\_et\_EDWIGE/neo4j/Rapport.pdf**