

Projet Neo4j

1- Scripts de définition de données

Création des entreprises :

```
// On crée 5 nœuds qui prennent le label Entreprise et contiennent 4 propriétés : nom, description et
// taille sont de type String. Secteur_activité est une liste de String car l'entreprise peut avoir
// plusieurs secteurs d'activité
```

```
CREATE (:Entreprise {Nom: "Atos" , Secteur_activité:["Technologies de l'information"], Description
: " Atos est une entreprise de services du numérique française, créée en 1997", Taille : " grande
entreprise"})
```

```
CREATE (:Entreprise {Nom: "Thales" , Secteur_activité:["l'aérospatiale", "la défense", "la sécurité",
"le transport terrestre "], Description : " Thales est un groupe d'électronique spécialisé dans
l'aérospatiale, la défense, la sécurité et le transport terrestre", Taille : "grande entreprise"})
```

```
CREATE (:Entreprise {Nom: "Danone" , Secteu_activité:["Agroalimentaire "], Description : " Danone
est une multinationale alimentaire française qui a été fondée à Barcelone et dont le siège social est à
Paris", Taille : "grande entreprise"})
```

```
CREATE (:Entreprise {Nom: "Orange" , Secteur_activité:[" Télécommunication "], Description : "
Orange est une entreprise française de télécommunications", Taille : "grande entreprise"})
```

```
CREATE (:Entreprise {Nom: "Samsung" , Secteur_activité:[" Électronique", "construction navale",
"bâtiment", "génie civil "], Description : " Le Groupe Samsung est un des principaux conglomérats
coréens.", Taille : "grande entreprise"})
```

Création des utilisateurs :

```
// On crée 4 nœuds qui prennent le label Utilisateur et contiennent 3 propriétés : nom, prénom et
//taille sont de type String. Description est une liste de String
```

```
CREATE (:Utilisateur {Nom: "Khadrouf" , Prénom:"Oumaima", Description : "élève en 3ème année à
tse ", Compétences : ["java", "c", "python", "R"]})
```

```
CREATE (:Utilisateur {Nom: "Birouk" , Prénom:"Yassine", Description : "consultant BI" ,
Compétences : ["java", "Pl/sql", "sql", "SAP"]})
```

```
CREATE (:Utilisateur {Nom: "Houmair" , Prénom:"Abdelaziz", Description : " Consultant Architecte
Cybersécurité" , Compétences : ["protocoles LAN", " VPN IPSEC", "python", "PKI"]})
```

```
CREATE (:Utilisateur {Nom: "EL gaf" , Prénom:"Oussama", Description : " Développeur Full Stack"
, Compétences : ["java", "javascript", "c", "c++", "php" ]})
```

Création des relations Utilisateur-Entreprise :

```
// Pour créer la relation on cherche d'abord les nœuds concernés cad le nœud avec le label utilisateur
// et le nœud avec le label entreprise puis on les stocke dans des variables. Enfin, on crée une relation
// travaille_pour entre une variable utilisateur et une ou plusieurs variables entreprise. Cette relation
// prend 3 propriétés : Statut de type String, Début de type date (YYYYMMDD) et Fin prend la valeur
// String "N/A" si l'utilisateur travaille toujours dans l'entreprise sinon elle prend une valeur de type
// date.
```

```
// On a adopté le type date pour le Début et la fin dans une perspective d'automatiser la création des
//relations a_travaillé_avec en fonction de ces dates et des relations des utilisateurs avec les
//entreprises
```

```
MATCH (u :Utilisateur { Nom: "Khadrouf" , Prénom: "Oumaima", Description : "élève en 3ème année
à tse "})
```

```
MATCH(e :Entreprise {Nom: "Samsung"})
```

```
CREATE (u)-[:travaille_pour{Début: 20180601, Fin: 20180830, Statut : "Stagiaire"}]->(e)
```

```
MATCH (u :Utilisateur {Nom: "Birouk" , Prénom: "Yassine", Description : "consultant BI"})
```

```
MATCH(e :Entreprise {Nom: "Atos"})
```

```
CREATE (u)-[:travaille_pour{Début: 20180901, Fin: "N/A", Statut : "Salarié"}]->(e)
```

```
MATCH (u :Utilisateur { Nom: "EL gaf" , Prénom: "Oussama", Description : " Développeur Full Stack" })
```

```
MATCH(e :Entreprise {Nom: "Danone"})
```

```
MATCH(e2 :Entreprise {Nom: "Thales"})
```

```
CREATE (u)-[:travaille_pour{Début: 20180101, Fin: "N/A", Statut : "Salarié"}]->(e)
```

```
CREATE (u)-[:travaille_pour{Début: 20170101, Fin: 20171130, Statut : "Salarié"}]->(e2)
```

```
MATCH (u :Utilisateur { Nom: "Houmair" , Prénom:"Abdelaziz", Description : " Consultant Architecte
Cybersécurité"})
```

```
MATCH(e :Entreprise {Nom: "Orange"})
```

```
MATCH(e2 :Entreprise {Nom: "Danone"})
```

```
CREATE (u)-[:travaille_pour{Début: 20180601, Fin:"N/A", Statut : "Salarié"}]->(e)
```

```
CREATE (u)-[:travaille_pour{Début: 20170201, Fin: 20180530, Statut : "Salarié"}]->(e2)
```

Création des relations Utilisateur-Utilisateur :

```
// D'abord on recherche les nœuds utilisateurs et on les stocke dans des variables, ensuite on crée une
// relation connaît ou a_travaillé_avec entre les nœuds.
```

```

MATCH (u :Utilisateur { Nom: "Khadrouf" , Prénom:"Oumaima", Description : "élève en 3ème année à tse "})
MATCH (u1 :Utilisateur {Nom: "Birouk" , Prénom: "Yassine", Description : "consultant BI"})
MATCH (u2 :Utilisateur { Nom: "Houmair" , Prénom: "Abdelaziz", Description : " Consultant Architecte Cybersécurité"})
MATCH (u3 :Utilisateur { Nom: "EL gaf" , Prénom: "Oussama", Description : " Développeur Full Stack" })
CREATE (u)-[:connait]->(u1)
CREATE (u)-[:connait]->(u2)
CREATE (u2)-[:connait]->(u1)
CREATE (u2)-[:connait]->(u3)
CREATE (u2)-[:a_travaillé_avec]->(u3)

```

2- Scripts de manipulation de données

Recherche d'utilisateurs

// Pour Optimiser la recherche d'utilisateurs et d'entreprises par nom on créer des indexes

```
CREATE INDEX ON :Utilisateur(Nom, Prénom)
```

```
CREATE INDEX ON :Entreprise(Nom)
```

// Les requêtes de recherches par nom

```
MATCH (u :Utilisateur { Nom: "Khadrouf" , Prénom: "Oumaima"}) RETURN u
```

```
MATCH (e :Entreprise { Nom: "Atos"}) RETURN e
```

Suggestions d'utilisateurs

// Pour chercher les utilisateurs qui ont travaillé avec un utilisateur x dans une entreprise y, on utilise la relation a_travaillé_avec pour chercher les utilisateurs et puis la relation travaille_pour pour verifier s'ils ont travaillé dans l'entreprise y

```
MATCH (:Utilisateur { Nom: "Houmair" , Prénom: "Abdelaziz"})-[:a_travaillé_avec]->(u :Utilisateur)-[:travaille_pour]->(e :Entreprise) WHERE e.Nom = "Danone" RETURN u
```

// Pour chercher les utilisateurs connus par les connaissances d'un utilisateur donné on utilise la //relation connait

```
MATCH (:Utilisateur { Nom: "Khadrouf" , Prénom: "Oumaima"})-[:connait]->( :Utilisateur)-[:connait]->(u :Utilisateur) RETURN u
```