

Rapport technique : Intégration de données étudiantes dans Neo4j

Aya hassan, Yassir tajmouati

juin 2025

Introduction

Ce rapport décrit les étapes d'intégration et de modélisation d'un jeu de données étudiantes (100 000 lignes) dans une base de données graphe Neo4j. Le traitement initial est effectué sur Google Colab afin de préparer les données, puis l'importation et la modélisation sont réalisées via Cypher dans Neo4j Desktop.

1 Préparation des données sur Google Colab

1.1 Chargement du fichier principal

```
import pandas as pd
df = pd.read_csv("students.csv")
```

1.2 Découpage en lots

```
batch_size = 10000
for i in range(0, len(df), batch_size):
    df.iloc[i:i+batch_size].to_csv(f"batch_{i//batch_size}.csv",
                                    index=False)
```

1.3 Fichier des semestres et statuts

```
df_semesters = df[['ID', 'S1', 'S2', 'S3', 'S4', 'S5', 'S6',
                    'S7', 'S8', 'S9', 'S10', 'S11', 'S12',
                    'Current_Status', 'Status_Description', '
                    Graduated']]
for i in range(0, len(df_semesters), batch_size):
    df_semesters.iloc[i:i+batch_size].to_csv(f"semesters_{i//
        batch_size}.csv", index=False)
```

2 Modélisation dans Neo4j Desktop

2.1 Création du projet

Un projet local nommé `students_graph` a été créé dans Neo4j Desktop. Les fichiers CSV ont été placés dans le dossier `import/` de la base.

2.2 Importation des étudiants

```
LOAD CSV WITH HEADERS FROM 'file:///batch_0.csv' AS row
MERGE (e:Etudiant {id: toInteger(row.ID)})
SET
    e.name = row.Name,
    e.gender = row.Gender,
    e.nationality = row.Nationality,
    e.birthdate = row.Birth_Date,
    e.start_year = toInteger(row.Start_Year),
    e.mark = toFloat(row.Mark),
    e.scholarship = toBoolean(row.Scholarship)
MERGE (v:Ville {nom: row.City})
MERGE (e)-[:VIENT_DE]->(v)
MERGE (b:Bac {type: row.Baccalaureat_Type})
MERGE (e)-[:A_BAC]->(b)
MERGE (ecole:Ecole {nom: row.School})
MERGE (e)-[:APPARTIENT_A]->(ecole)
MERGE (s:Specialite {nom: row.Specialty})
MERGE (e)-[:ETUDIE]->(s)
```

2.3 Importation des semestres (ex : batch 9)

```
LOAD CSV WITH HEADERS FROM 'file:///semesters_9.csv' AS row
MATCH (e:Etudiant {id: toInteger(row.ID)})
SET
    e.s1 = row.S1,
    e.s2 = row.S2,
    e.s3 = row.S3,
    e.s4 = row.S4,
    e.s5 = row.S5,
    e.s6 = row.S6,
    e.s7 = row.S7,
    e.s8 = row.S8,
    e.s9 = row.S9,
    e.s10 = row.S10,
    e.s11 = row.S11,
    e.s12 = row.S12,
    e.status = row.Current_Status,
    e.status_description = row.Status_Description,
    e.graduated = toBoolean(row.Graduated)
```

3 Analyse et Visualisation du Graphe

3.1 Requêtes de vérification

```
MATCH (e:Etudiant) RETURN count(e);
```

Listing 1 – Nombre total d'étudiants

```
MATCH (e:Etudiant)-[:ETUDIE]->(s:Specialite),  
      (e)-[:VIENT_DE]->(v:Ville)  
RETURN e.name, s.nom AS specialite, v.nom AS ville  
LIMIT 10;
```

Listing 2 – Étudiants avec spécialité et ville

3.2 Visualisation du graphe dans Neo4j Browser

Pour afficher un sous-ensemble du graphe :

```
MATCH (e:Etudiant)-[r]->(n)  
RETURN e, r, n  
LIMIT 100;
```

Cliquez ensuite sur l'icône "Graphe" pour voir la visualisation.

3.3 Analyse avancée (exemples)

```
MATCH (e:Etudiant)-[:ETUDIE]->(s:Specialite)  
RETURN s.nom AS specialite, count(e) AS nb_etudiants  
ORDER BY nb_etudiants DESC  
LIMIT 10;
```

Listing 3 – Spécialités les plus fréquentes

```
MATCH (e:Etudiant)  
WHERE e.s1 = "validate" AND e.s2 = "validate" AND e.s3 = "  
      validate"  
      AND e.s4 = "validate" AND e.s5 = "validate" AND e.s6 = "  
      validate"  
      AND e.s7 = "validate" AND e.s8 = "validate" AND e.s9 = "  
      validate"  
      AND e.s10 = "validate" AND e.s11 = "validate" AND e.s12 = "  
      validate"  
RETURN e.name, e.graduated;
```

Listing 4 – Validation complète des semestres

4 Conclusion

La base de données graphe a été construite avec succès à partir de 100 000 étudiants. Toutes les relations académiques ont été modélisées dans Neo4j et les données sont prêtes

pour la visualisation, l'analyse ou l'intégration dans une application web ou un dashboard interactif.