

Neo4j

Projet Neo4j : Analyse de la base de données

Requêtes de création de la base de données

Création des nœuds Lieu, Accident et de la relation SE_DEROULE_A

```
LOAD CSV WITH HEADERS FROM 'file:///accidents.csv' AS row
FIELDTERMINATOR ';'
MERGE (l:Lieu {latitude: toFloat(replace(row.lat, '"', '')), longitude: toFloat(replace(row.long, '"', ''))})
SET l.voie = row.adr,

    `l.intersection = toInteger(row.int)`

MERGE (a:Accident {num_acc: toInteger(row.Num_Acc)})
SET a.type_collision = toInteger(row.col),

    `a.jour = toInteger(row.jour),`

    `a.mois = toInteger(row.mois),`

    `a.annee = toInteger(row.an),`

    `a.heure = row.hrmn,`

    `a.commune = toInteger(row.com),`

    `a.departement = toInteger(row.dep),`

    `a.agglomeration = toInteger(row.agg)`

MERGE (a)-
[:SE_DEROULE_A {climat: toInteger(row.atm), temps: toInteger(row.lum)}]->(l);
```

Création du nœud Vehicule et la relation EST_IMPLIQUE_DANS (entre Vehicule et Accident)

```
LOAD CSV WITH HEADERS FROM 'file:///vehicules.csv' AS row
FIELDTERMINATOR ';'
MERGE (a:Accident {num_acc: toInteger(row.Num_Acc)})

MERGE (v: Vehicule {id_vehicule: toInteger(row.id_vehicule)})
SET v.categorie_vehicule = toInteger(row.catv),

    v.obstacle_fixe_heurt = toInteger(row.obs),

    v.obstacle_mobile_heurt = toInteger(row.obsm),

    v.choc = toInteger(row.choc),

    v.moteur = toInteger(row.motor),

    v.manoeuvre = toInteger(row.manv)

MERGE (v)-[:EST_IMPLIQUE_DANS]->(a);
```

Création du nœud Usager et des relations A_FAIT_UN (pour le nœud Accident) et UTILISE (pour le nœud Vehicule)

```
LOAD CSV WITH HEADERS FROM 'file:///usagers.csv' AS row
FIELDTERMINATOR ';'
MERGE (a: Accident {num_acc: toInteger(row.Num_Acc)})
MERGE (v: Vehicule {id_vehicule: toInteger(row.id_vehicule)})

MERGE (u: Usager {id_usager: toInteger(row.id_usager)})
SET u.type_usager = toInteger(row.catu),

    u.sexe = toInteger(row.sexe),

    u.gravite = toInteger(row.grav),

    u.age = toInteger(row.an_nais),
```

```
u.siege = toInteger(row.place),

u.trajet = toInteger(row.trajet)

MERGE (u)-[:A_FAIT_UN]->(a)
MERGE (u)-[:UTILISE]->(v);
```

Requêtes d'analyse de la base de données

1. Top 20 des départements qui enregistrent les chiffres les plus élevés en termes d'accidents graves

```
MATCH (a:Accident)<-[:A_FAIT_UN]-(u:Usager),

      (a)-[:SE_DEROULE_A]->(l:Lieu)

WHERE u.type_usager = 3 AND a.departement IS NOT NULL AND u.gravite IN [2, 3]

RETURN a.departement AS departement,

       u.type_usager AS type_usager,

       COUNT(a) AS nb_accidents_graves

ORDER BY nb_accidents_graves DESC

LIMIT 20
```

2. Le nombre d'accidents par nombre de véhicules impliqués et par niveau de gravité

```
MATCH (a:Accident)<-[:EST_IMPLIQUE_DANS]-(v:Vehicule), (u:Usager)-
[:A_FAIT_UN]->(a),

(u)-[:UTILISE]->(v)

WITH a, COUNT(v) AS nombre_vehicules

WHERE nombre_vehicules > 1

MATCH (a)<-[:A_FAIT_UN]-(u:Usager)
```

```

RETURN nombre_vehicules AS vehicules_impliquees,

        u.gravite AS gravite,

        COUNT(a) AS nb_accidents

ORDER BY nb_accidents DESC

```

3. Le nombre d'accidents selon le type de trajet, la voie, l'heure et la gravité

```

MATCH (a:Accident)<-[:A_FAIT_UN]-(u:Usager), (a)-[:SE_DEROULE_A]->(l:Lieu)

WHERE u.trajet IS NOT NULL AND u.gravite IS NOT NULL

RETURN l.voie AS voie, u.trajet AS type_trajet, a.heure AS heure,

        u.gravite AS gravite,

        COUNT(a) AS nb_accidents

ORDER BY nb_accidents DESC

```

4. Le nombre d'accidents selon type de véhicule et le lieu de l'accident

```

MATCH (a:Accident)-[:SE_DEROULE_A]->(l:Lieu),

        (a)<-[:EST_IMPLIQUE_DANS]-(v:Vehicule)

RETURN l.voie AS voie, l.latitude AS latitude, l.longitude AS longitude, v.categorie_vehicule AS type_vehicule, COUNT(a) AS nb_accidents

ORDER BY mois, nb_accidents DESC

LIMIT 20

```

5. Le nombre d'accidents selon le sexe , le lieu et la gravité

```
MATCH (a:Accident)<-[:A_FAIT_UN]-(u:Usager),
```

```
(a)-[:SE_DEROULE_A]->(l:Lieu)
```

```
RETURN l.voie AS voie, l.latitude AS latitude, l.longitude AS longitude, u.sexe AS sexe, u.gravite AS gravite, COUNT(a) AS nb_accidents
```

```
ORDER BY nb_accidents DESC
```

```
LIMIT 20
```