Fondamentaux des bases de données NoSQL
Miniprojet

28 Février 2025

YERIMA Sika
41001058
Master 1- CMI D3S

Introduction

Pour ce mini projet porté sur l'utilisation de l'apprentissage de MongoDB, j'ai sélectionné sur Kaggle, un jeu de données qui recense les retours de produits effectués aux États-Unis par les agences responsables de la sûreté des produits alimentaires.
Nous avons une liste de produits ayant été demandés au rappel, ainsi que la marque en question et la date où la notification a été émise.

Ainsi, j'ai décidé d'examiner les différents produits n'adhérant pas aux normes de sécurité, quelles sont les causes les plus courantes de rappels et combien d'États sont impactés, et surtout examiner dans quelle mesure la sécurité du consommateur est respectée.

Dans un contexte où la sécurité alimentaire est une préoccupation majeure, l'analyse des rappels de produits permet d'identifier les tendances et d'améliorer les protocoles de contrôle qualité.
Ce projet permettra non seulement de se familiariser avec MongoDB, mais aussi de mettre en pratique des techniques d'exploration et d'analyse de données appliquées à un enjeu concret de santé publique.

À cet effet, j'ai créé une base de données du nom de mini projet. Base contenant **la collection "recalls"** pour rappel/retour du produit.

Nous allons en premier lieu interroger la base MongoDB en utilisant des requêtes find avec différents filtres et projections afin d'extraire des informations pertinentes. Ensuite, nous mettrons à jour la collection en insérant, modifiant et supprimant des documents. Enfin, nous exploiterons les fonctionnalités d'agrégation et MapReduce pour analyser les tendances des rappels alimentaires et obtenir des insights sur la sécurité des produits.

Une dernière partie pourra être consacrée à un peu d'explorations sous neo4.j.

Variable	Signification	Type
_id	Identification du rappel	Code
title	Annonce de retour	Chaîne de caractères
company_annonce_dttm	Retour du produit par la compagnie	Date
notification_dttm	Retour demandé par l'agence	Date
recall_reason	Raison de la demande de retour du produit	Chaîne de caractères
company_name	La compagnie responsable du produit	Chaîne de caractères
brand_name	La marque du produit	Chaîne de caractères
product_description	Quel produit en question	Chaîne de caractères
impacted_state	États impactés par le rappel	Liste de chaîne de caractères
agency	L'agence nationale responsable du rappel	Chaîne de caractères
risk_level	Niveau de risque du produit	Chaîne de caractères

Table 1: Tableau des variables d'intérêt

Nous allons à présent débuter l'analyse.

1 Interroger la base (avec la fonction "find")

1.1 Un filtre simple (sans opérateur de comparaison)

1. Nous voulons savoir quel(s) produit(s) de la fameuse marque de chips "Lay's" ont été rappelés et pourquoi ?

```
db.recalls.find({brand_name: "Lay's " },{'_id': 0, 'recall_reason':1,'
    product_description':1})
```

.

1.2 Un filtre sur liste (en utilisant $elemMatch) si la collection contient une liste

1. Quel(s) produit(s) a (ont) été rappelé(s) dans l'état de New York ('NY') ?

```
db.recalls.find({ impacted_states: { $elemMatch: { $eq: 'NY' } } },{'_id': 0,'
    product_description':1,'brand_name':1})
```

.

1.3 Deux filtres sur des paires (clé, valeur) imbriquées (notation pointée)

1.Quel(s) produit(s) a (ont) été rappelé(s) de l'état du New Jersey (NJ) et pourquoi ?

```
db.recalls.find({ "impacted_states.0":'NJ'},
{'_id': 0,'brand_name':1,'product_description':1,'impacted_states':1 ,'recall_reason'
    :1})
```

.
2.Quel(s) est (sont) le(s) produit(s) ayant été rappelé(s) dans l'état de Californie ('CA'), dans l'état du Delaware ('DE') ou de Rhode Island ('RI'); qui (a) ont un risque de niveau identifié ?

```
db.recalls.find({
    $and:[
    { "impacted_states.0": {$in:[ "CA",'DE','RI'] }},
    {risk_level:{$exists:true, $ne: null }}]},
{'_id': 0,'product_description':1,'risk_level':1})
```

.

1.4 Deux filtres avec des opérateurs de comparaison

1.Quels produits ont été soumis au rappel depuis le début de cette année ?

```
db.recalls.find({
notification_dttm: { $gt: ("2024-12-31T05:00:00+00:00") }
},{'_id':0,'brand_name':1,'product_description':1,'notification_dttm':1})
```

.
2.Quels produits ont été rappelés durant le mois de décembre 2024 et quand l'agence a t-elle demandé le retour de ces produits ?

```
db.recalls.find({ company_announce_dttm: { $gt: ("2024-12-01T05:00:00+00:00") ,$lt: (
    "2024-12-31T05:00:00+00:00") } },
{'_id': 0,'product_description':1,'notification_dttm':1})
```

.

1.5 Deux filtres complexes avec au moins deux composantes des propositions ci-dessus.

1.Quels produits ont été rappelés depuis 2025 par la FDA et quand le rappel a-t-il été notifié ?

```
db.recalls.find({
  $and: [
    { notification_dttm: { $gte:("2025-01-01T05:00:00+00:00") } },
    { agency: "FDA" } ]
},{'_id': 0,'brand_name':1,'product_description':1,'notification_dttm':1})
```

.
2.Quels sont les produits de marque connus qui ont été rappelés pour des raisons de contaminations ou de maladies ?

```
db.recalls.find({
    $and:[
    {brand_name:{$ne:null}},
  {$or: [
    { recall_reason: { $regex: "Contamination" }},
    { recall_reason: { $regex: "Illness" } } ]}]
},{'_id': 0,'brand_name':1,'product_description':1,'recall_reason':1})
```

.

2 Mettre à jour la collection

2.1 Une requête pour ajouter ou modifier une paire (clé, valeur) (Update/Set)

1.Remplacer la valeur null qui était dans brand_name, company_name et product_description par la marque, la compagnie et le produit en question.

```
db.recalls.updateOne(
  { title: { $regex:'DJs_Boudain_LLC_Recalls_Sausage_Link_Products__Due_to_Possible_
     Foreign_Matter_Contamination'}},
  { $set: { company_name: 'DJs_Boudain,LLC',brand_name:'DJs_Boudain',
     product_description:'Sausage'} })
```

Vérification

```
db.recalls.find({ brand_name:'DJs_Boudain'})
```

.

2.2 Une requête pour supprimer une paire clé/valeur (Update/unset)

1.Supprimer les raisons pour les produits, dont le nom de la compagnie ou le nom de la marque n'existe pas.

```
db.recalls.updateMany({ $or: [
  { brand_name: null},{company_name: null}]},
  { $unset: { recall_reason: "" } }
```

Vérification

```
db.recalls.find({ $or: [
  { brand_name: null },{company_name: null}]})
```

.

2.3 Une requête pour ajouter un document au choix (Insert)

1. Ajouter un produit à rappeler.

```
  db.recalls.insertOne({
  product_description: "Super Chips",
  company_name:"Snack Co, LLC",
  brand_name: "Snack Co",
  notification_dttm: ("2025-02-15T05:00:00+00:00"),
  recall_reason: "Salmonella contamination",
  agency: "FDA",
  impacted_states: Array (2),
  risk_level:"High class"
})
db.recalls.updateOne({company_name:"Snack Co, LLC"},{
  $set: {impacted_states:['LA','TX']}})
```

Vérification

```
  db.recalls.find({company_name:"Snack Co, LLC"})
```

.

2.4 Une requête pour ajouter un élément au choix dans une liste ($push)

1.D'autres états sont concernés par le rappel des 'Super chip's de "Snack Co, LLC".

```
  db.recalls.updateOne({$and:[{product_description: "Super Chips"},
  { company_name:"Snack Co, LLC" }]},
```

```
{ $push:{impacted_states: { $each: ['FR', 'DE', 'CA','DA']}}})
```

Vérification

```
db.recalls.find({company_name:"Snack Co, LLC" },{impacted_states:1})
```

.

2.5 Une requête pour supprimer un document ou plusieurs au choix

1.je veux retirer les documents où il n'y a pas de nom de compagnie ou de raison de retour.

```
db.recalls.deleteMany({ $or: [{ brand_name: null }, { recall_reason: null }] })
```

Vérification

```
db.recalls.find({ $or: [{ brand_name: null }, { recall_reason: null }] })
```

.
2.je veux retirer le rappel du Shatavari Powder.

```
db.recalls.deleteOne({product_description:"Shatavari Powder"})
```

Vérification

```
db.recalls.find({product_description:"Shatavari Powder"})
```

.

3 Agrégations et MapReduce

3.1 Deux requêtes d'agrégation simples (pipeline) sans synthèse (sans fonction d'agrégation)

1. On veut voir tous les produits rappelés pour raison de salmonella.

```
db.recalls.aggregate([
{ $match: { recall_reason: {$regex:"Salmonella" } }}
])
```

.
2. Produits rappelés dans au moins 3 États différents.

```
db.recalls.aggregate([
  {
    $project: {
      brand_name: 1, product_description: 1, impacted_states: 1,
      total_states: { $size: "$impacted_states" } }
  },
  { $match: { total_states: { $gte: 3 } } },
  { $sort: { total_states: -1 } }
]);
```

.

3.2 Requêtes avec pipeline d'agrégation impliquant au moins des opérateurs et une fonction de synthèse

1. Combien de produits ont été rappelés pour raison de contamination ?

```
  db.recalls.aggregate([
{$match: { recall_reason: {$regex:("contamination") }}},
  { $group: { _id: "$recall_reason",
  total_recalls: { $sum: 1 } } }
  ])
```

.
2.Combien de rappel il y a-t-il eu par après 2023 ?

```
db.recalls.aggregate([
  { $match: { notification_dttm: { $gt:("2023-01-01T05:00:00+00:00") }
   }
  },
  { $group: { _id: { $substr: ["$notification_dttm", 0, 4]}, total: { $sum: 1 }
  } }
  ])
```

.
3. Quel est le dernier rappel qui a été consigné ?

```
db.recalls.aggregate([
  {$group: {
      _id: null,
      first_recall: { $min: "$notification_dttm" },
      last_recall: { $max: "$notification_dttm" }}
  }
  ]);
```

.
4.Quelles sont les 5 raisons les plus fréquentes dans les rappels ?

```
db.recalls.aggregate([
  { $unwind: "$recall_reason" },
  { $group: { _id: "$recall_reason", count: { $sum: 1 } } },
  { $sort: { count: -1 } },
  { $limit: 5 }
  ])
```

.
5. Lister tous les fabricants impliqués dans des rappels récents (depuis Septembre 2024) et les produits en question.

```
db.recalls.aggregate([
{ $match: { notification_dttm: {$gte:("2024-09-01T05:00:00+00:00") }}
},
  { $group: { _id: "$brand_name",
  recalled_products: { $push: "$product_description" } } } ])
```

.
6. Nombre de rappels par agence responsable.

```
  db.recalls.aggregate([
  {$group: {
      _id: "$agency",
      total_recalls: { $sum: 1 }}}
      ]);
```

.
7.Liste des 20 États les plus impactés (impacted_states).

```
db.recalls.aggregate([
  { $unwind: "$impacted_states" },
  { $group: {
      _id: "$impacted_states",
      total_recalls: { $sum: 1 }
    }},
  { $sort: { total_recalls: -1 } },
  { $limit: 20 }
  ]);
```

.

3.3 Reprendre les requêtes de synthèse avec les fonctions Map/Reduce.

1. Nombre total de rappels par fabricant.

```
  db.recalls.mapReduce(
  function()
  { emit(this.company_name, 1); },
  function(key, values) { return Array.sum(values); },
  { out: "recalls_by_manufacturer" }
);
```

.

2. Trouver le dernier rappel consigné.

```
  db.recalls.mapReduce(
  function()
  { emit("last_recall", this.notification_dttm; },
  function(key, values) { return values.sort().pop(); },
  { out: "last_recall" }
);
```

.

3. Nombre de rappels par agent responsable.

```
  db.recalls.mapReduce(
  function() { emit(this.agency, 1);
  },
  function(key, values) { return Array.sum(values); },
  { out: "recalls_by_agency" }
);
```

.

4. Liste des 20 États les plus impactés(impacted_states).

```
db.recalls.mapReduce(
  function()
  {
    if (this.impacted_states) {
      this.impacted_states.forEach(state => emit(state, 1));}
  },
  function(key, values) {return Array.sum(values);},
  {out: { inline: 1 },
    query: {},
    sort: { value: -1 },
    limit: 20});
```

.

5. Nombre de rappels contenant "Contamination" ou "Illness".

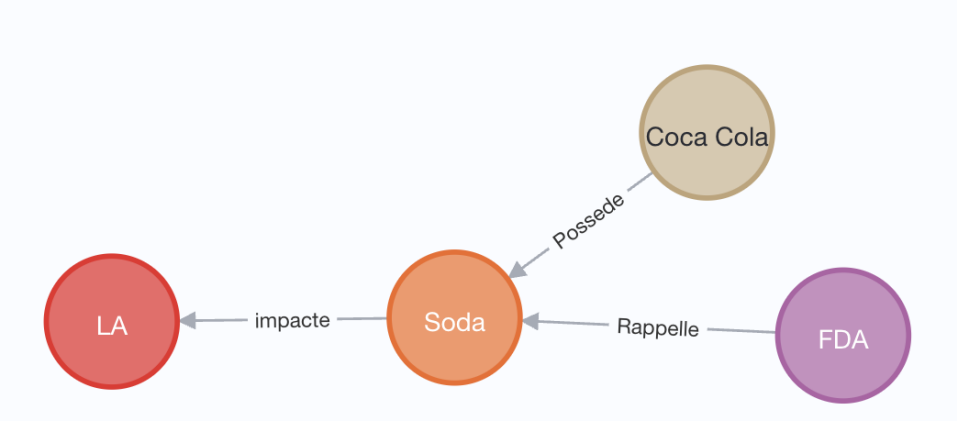
```
db.recalls.mapReduce(
  function() {
    if (this.recall_reason && (this.recall_reason.includes("Contamination") || this.
        recall_reason.includes("Illness"))) {
      emit("affected_recalls", 1);}},
  function(key, values) { return Array.sum(values); },
  { out: "contaminated_recalls" });
```

4 Bonus: Graphe des rappels par Neo4j et exploration

4.1 Création des relations

On peut modéliser des relations complexes de cette collection de façon intuitive et efficace entre les produits, les marques et les rappels en créant des relations avec des attributs pertinents.
Par exemple, le lien entre une agence et le produit dont elle demande le rappel.

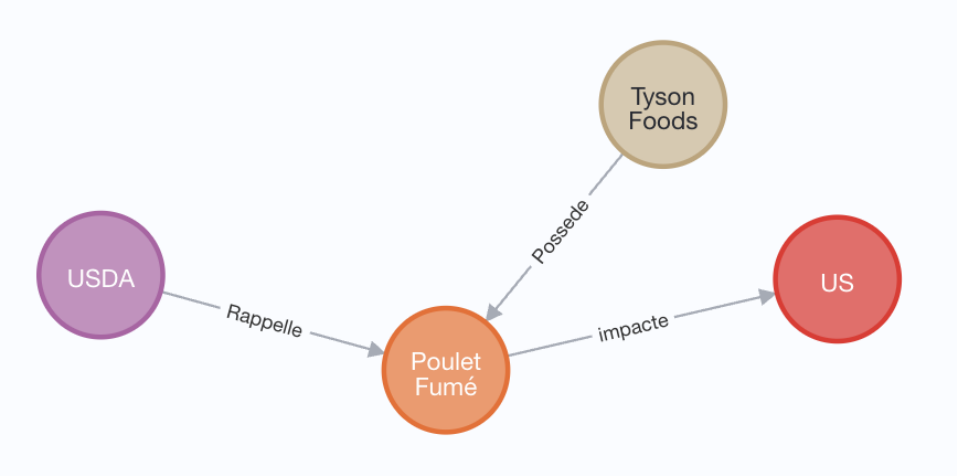
```
CREATE (a1:Agency {Nom: "FDA"})
CREATE (p1:Produit {Nom: "Soda", Marque: "Coca_Cola"})
CREATE (a1)-[:Rappelle {Quand: date("2025-02-20"), Pourquoi:"Trop_de_sucre"}]->(p1)
CREATE (c1:Company {Nom: "Coca_Cola"})
CREATE (c1)-[:Possede]->(p1)
CREATE (s1:Pays{Code:"LA"})
CREATE (p1)-[:impacte {NiveauRisque:"Substantiel"}]->(s1)
```



Voici comment "la FDA qui rappelle un soda de la compagnie Coca Cola se présente le 20 Février 2025" et qui impacte l'État de Los Angeles 'LA' de manière substantielle.

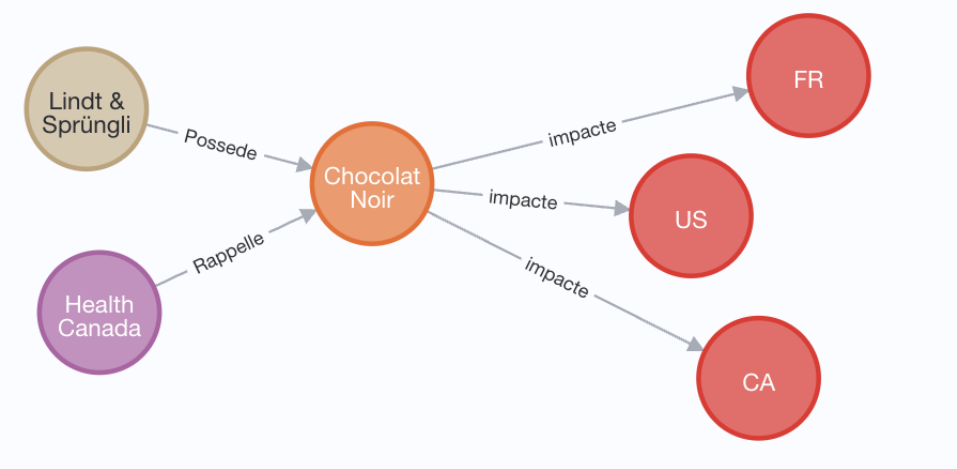
Voici un autre exemple d'addition de produit à l'ensemble des retours.

```
CREATE (a2:Agency {Nom: "USDA"})
CREATE (p2:Produit {Nom: "Poulet_Fum", Marque: "Tyson"})
CREATE (a2)-[:Rappelle {Quand: date("2025-01-15"), Pourquoi:"Prsence_de_salmonelle"
    }]->(p2)
CREATE (c2:Company {Nom: "Tyson_Foods"})
CREATE (c2)-[:Possede]->(p2)
CREATE (s2:Pays {Code: "US"})
CREATE (p2)-[:impacte {NiveauRisque:"Critique"}]->(s2)
```



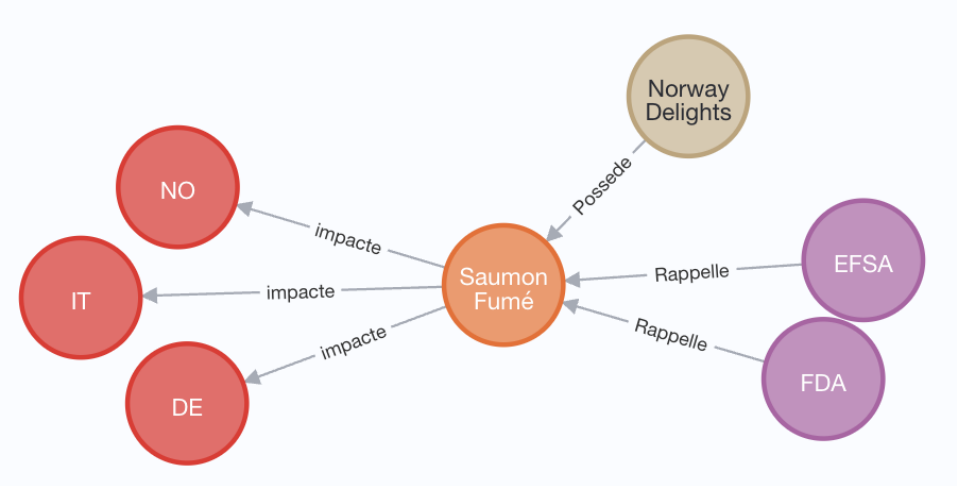
On a également le cas où on nécessite un rappel d'un autre produit dans plusieurs pays.

```
CREATE (a3:Agency {Nom: "Health_Canada"})
CREATE (p3:Produit {Nom: "Chocolat_Noir", Marque: "Lindt"})
CREATE (a3)-[:Rappelle {Quand: date("2025-03-10"), Pourquoi:"Traces_de_lait_non_
    dclar"}]->(p3)
CREATE (c3:Company {Nom: "Lindt_&_Sprngli"})
CREATE (c3)-[:Possede]->(p3)
CREATE (s3:Pays {Code: "CA"})
CREATE (s4:Pays {Code: "US"})
CREATE (s5:Pays {Code: "FR"})
CREATE (p3)-[:impacte {NiveauRisque:"Modr"}]->(s3)
CREATE (p3)-[:impacte {NiveauRisque:"Modr"}]->(s4)
CREATE (p3)-[:impacte {NiveauRisque:"lev"}]->(s5)
```



.
Maintenant un produit rappelé par plusieurs agences.

```
CREATE (a4:Agency {Nom: "FDA"})
CREATE (a5:Agency {Nom: "EFSA"})
CREATE (p4:Produit {Nom: "Saumon_Fum", Marque: "Norway_Delights"})
CREATE (a4)-[:Rappelle {Quand: date("2025-04-05"), Pourquoi:"Listeria_monocytogenes"
    }]->(p4)
CREATE (a5)-[:Rappelle {Quand: date("2025-04-07"), Pourquoi:"Listeria_monocytogenes"
    }]->(p4)
CREATE (c4:Company {Nom: "Norway_Delights"})
CREATE (c4)-[:Possede]->(p4)
CREATE (s6:Pays {Code: "NO"})
CREATE (s7:Pays {Code: "DE"})
CREATE (s8:Pays {Code: "IT"})
CREATE (p4)-[:impacte {NiveauRisque:"Critique"}]->(s6)
CREATE (p4)-[:impacte {NiveauRisque:"lev"}]->(s7)
CREATE (p4)-[:impacte {NiveauRisque:"Modr"}]->(s8)
```



Enfin, une entreprise ayant plusieurs produits rappelés.

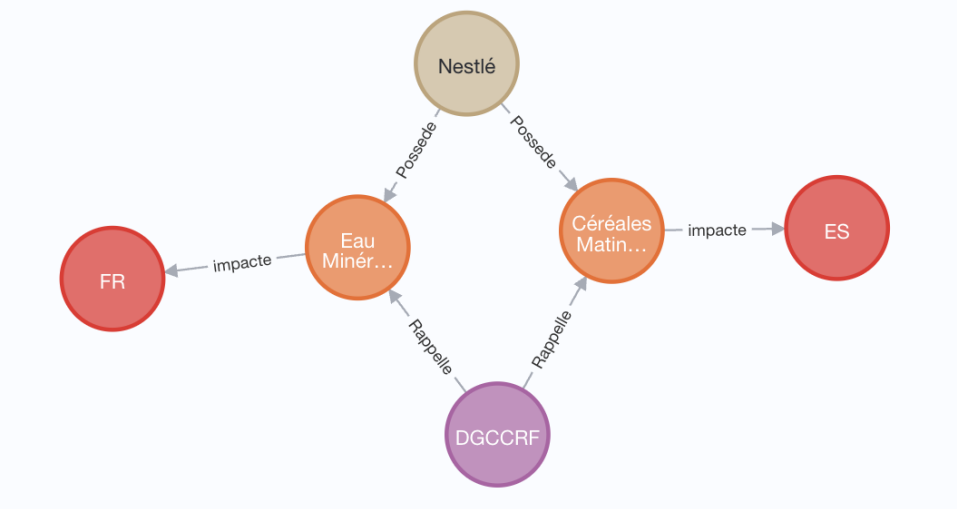
```
CREATE (c5:Company {Nom: "Nestl"})
CREATE (p5:Produit {Nom: "Eau Minrale", Marque: "Nestl Pure Life"})
CREATE (p6:Produit {Nom: "Crales Matinales", Marque: "Chocapic"})

CREATE (a6:Agency {Nom: "DGCCRF"})
CREATE (a6)-[:Rappelle {Quand: date("2025-05-12"), Pourquoi:"Contamination
    bactrienne"}]->(p5)
CREATE (a6)-[:Rappelle {Quand: date("2025-05-15"), Pourquoi:"Prsence de corps
    trangers"}]->(p6)

CREATE (c5)-[:Possede]->(p5)
CREATE (c5)-[:Possede]->(p6)

CREATE (s9:Pays {Code: "FR"})
CREATE (s10:Pays {Code: "ES"})

CREATE (p5)-[:impacte {NiveauRisque:"Modr"}]->(s9)
CREATE (p6)-[:impacte {NiveauRisque:"lev"}]->(s10)
```



4.2 Analyse des connexions

1. Lister tous les rappels de produits et leurs raisons.

```
MATCH (p:Produit)<-[r:Rappelle]-(a:Agency)
RETURN a.Nom AS Agence, p.Nom AS Produit, p.Marque AS Marque, r.Quand AS Date, r.
    Pourquoi AS Raison
ORDER BY r.Quand DESC
```

.
2. Trouver les produits rappelés par une agence spécifique.

```
MATCH (a:Agency)-[:Rappelle]->(p:Produit)
WHERE a.Nom = "FDA"
RETURN p.Nom AS Produit, p.Marque AS Marque
```

.
3. Lister toutes les entreprises et leurs produits rappelés.

```
MATCH (c:Company)-[:Possede]->(p:Produit)<-[:Rappelle]-(a:Agency)
RETURN c.Nom AS Entreprise, COLLECT(p.Nom) AS ProduitsRappeles, COUNT(p) AS
   NombreDeProduits
ORDER BY NombreDeProduits DESC
```

.
4. Identifier les pays les plus impactés par les rappels.

```
MATCH (p:Produit)-[i:impacte]->(s:Pays)
RETURN s.Code AS Pays, COUNT(p) AS NombreDeProduitsImpactes
ORDER BY NombreDeProduitsImpactes DESC
LIMIT 10
```

.
5. Trouver les produits rappelés qui présentent un risque critique.

```
MATCH (p:Produit)-[i:impacte {NiveauRisque: "Critique"}]->(s:Pays)
RETURN p.Nom AS Produit, p.Marque AS Marque, COLLECT(s.Code) AS PaysImpactes
```

.
6. Trouver les agences qui rappellent souvent des produits.

```
MATCH (a:Agency)-[:Rappelle]->(p:Produit)
RETURN a.Nom AS Agence, COUNT(p) AS NombreDeRappels
ORDER BY NombreDeRappels DESC
```

.
7. Trouver toutes les entreprises ayant des produits impactés en France.

```
MATCH (c:Company)-[:Possede]->(p:Produit)-[:impacte]->(s:Pays)
WHERE s.Code = "FR"
RETURN DISTINCT c.Nom AS Entreprise, COLLECT(p.Nom) AS ProduitsImpactes
```

.

Conclusion

En combinant MongoDB pour la gestion des données brutes et Neo4j pour l'analyse des relations, on obtient une approche puissante pour suivre et anticiper les tendances en matière de sécurité alimentaire. Ce type d'analyse peut aider les régulateurs et les entreprises à mieux réagir face aux crises et à améliorer la traçabilité des produits.