

École Nationale Supérieure d'Informatique, Alger
Module Base de Données Avancées

Rapport Travail Pratique BDDA

Réalisé par : Messar Cylia et Tagzirt Elissa
Spécialité : Systèmes Informatiques
Groupe : 1
2023-2024

Sommaire

Introduction (Page 3)

Résumé (Page 4 - 5)

Description des données (Page 6-8)

Base de données NoSQL (Page 9 -10)

L'outil MongoDB (Page 11 -13)

Réalisation des requêtes en MongoDB (Page 14 - 26)

Conclusion (Page 27)

Bibliographie (Page 28)

Introduction

Les bases de données sont un **élément essentiel** en informatique, permettant de manipuler et de gérer des volumes de données **importants**. Parmi les outils populaires et concepts répandus actuellement, les bases de données **NoSQL** se distinguent. Ces dernières ne visent pas à remplacer le SQL, mais à répondre à des besoins spécifiques en matière de gestion de données.

Dans ce travail, nous explorerons les pratiques liées aux bases de données NoSQL, en nous concentrant sur les requêtes et autres manipulations de données pré-traitées et converties au **format JSON**. Nous choisirons ensuite quelques requêtes stratégiques et préparerons un environnement (interfaces) pour répondre aux besoins fréquents en termes de requêtes de bases de données.

Résumé

Ce travail vise à explorer les données traitées lors du semestre précédent avec Monsieur Ait Yahia Ali Yacine. Le sujet portait sur l'augmentation de données issues d'études agronomiques. Nous nous intéressons particulièrement à deux fichiers essentiels: `archides_riz.csv` et `data_plants.csv`. Le premier contient des informations sur deux cultures, le riz et l'arachide, tandis que le second couvre la pomme de terre et le riz.

L'objectif est de déterminer la structure et le format de données les plus appropriés pour nos fichiers CSV en fonction de critères spécifiques. Nous choisirons ensuite l'outil le plus adapté (**MongoDB**), créerons la base de données, puis exposerons des requêtes stratégiques pour répondre aux besoins réels de notre étude.

Notre étude se concentre sur les besoins en eau pour différentes cultures selon les mois, ainsi que sur l'irrigation. Nous prenons également en compte d'autres paramètres:

- 1) **CropDays**: nombre de jours de culture (jours).
- 2) **Soil Moisture**: humidité du sol (centimètres cubes d'eau par centimètre cube de sol, cm^3/cm^3 ou pourcentage %).
- 3) **Soil Temperature**: température du sol (degrés Celsius, °C).
- 4) **Temperature**: température de l'air (degrés Celsius, °C).
- 5) **Humidity**: humidité relative (pourcentage %).
- 6) **Irrigation(Y/N)**: irrigation (Oui/Non).
- 7) **Water req**: besoin en eau (millimètres, mm).
- 8) **Month**: mois.
- 9) **Min Temp**: température minimale (degrés Celsius, °C).
- 10) **Max Temp**: température maximale (degrés Celsius, °C).
- 11) **Wind**: vitesse du vent (mètres par seconde, m/s).
- 12) **Sun**: ensoleillement (heures par jour, h/j).
- 13) **Rad**: Radiation solaire (mégajoules par mètre carré par jour, $\text{MJ}/\text{m}^2/\text{j}$).
- 14) **Rain**: précipitations (millimètres, mm).
- 15) **Altitude**: altitude (mètres, m).
- 16) **Latitude**: latitude (degrés).
- 17) **Longitude**: longitude (degrés).
- 18) **Crop**: type de culture.
- 19) **Soil**: type de sol.
- 20) **City**: ville.

Description des données

1. Le premier fichier (Arachides et Riz):

Le fichier contient des données sur deux types de cultures : le **riz** et l'**arachide**. Chaque entrée représente un ensemble de paramètres mesurés ou observés pour une culture spécifique sur une certaine période. Voici les détails des paramètres:

- **CropDays:** Nombre de jours de culture (jours)
- **Soil Moisture:** Humidité du sol (centimètres cubes d'eau par centimètre cube de sol, cm^3/cm^3)
- **Soil Temperature:** Température du sol (degrés Celsius, °C)
- **Temperature:** Température de l'air (degrés Celsius, °C)
- **Humidity:** Humidité relative (pourcentage %)
- **CropType:** Type de culture (riz ou arachide)
- **Irrigation(Y/N):** Indication si l'irrigation a été utilisée (Oui/Non)

Relations possibles entre ces paramètres :

a) *CropDays et CropType:*

Les différentes cultures peuvent avoir des durées de croissance différentes. Par exemple, le riz et l'arachide peuvent nécessiter un nombre différent de jours pour arriver à maturité.

b) *Soil Moisture et Irrigation(Y/N):*

L'irrigation influence directement l'humidité du sol. Les entrées avec "Irrigation: Y" devraient généralement montrer des valeurs plus élevées d'humidité du sol comparées à celles avec "Irrigation: N".

c) *Temperature (air) et Soil Temperature:*

La température de l'air et celle du sol sont souvent corrélées. Une hausse de la température de l'air peut entraîner une hausse de la température du sol.

d) *Humidity et Irrigation(Y/N):*

L'irrigation peut aussi avoir un effet sur l'humidité relative, surtout dans des environnements fermés comme les serres.

e) *CropType et Soil Moisture:*

Différentes cultures ont des besoins en eau différents, ce qui peut être reflété par les niveaux d'humidité du sol. Par exemple, le riz, ayant besoin de plus d'eau, pourrait montrer des valeurs plus élevées d'humidité du sol par rapport à l'arachide.

f) *CropDays et Soil Moisture:*

Au fur et à mesure que les jours de culture augmentent, l'humidité du sol peut diminuer si l'irrigation n'est pas suffisante pour compenser l'eau absorbée par les plantes.

2. Deuxième Fichier (Data Plants):

Le deuxième fichier contient des données détaillées sur les cultures de pomme de terre et de riz. Chaque entrée représente un ensemble de paramètres mesurés ou observés pour une culture spécifique durant un mois donné. Voici les détails des paramètres:

- **Water req:** Besoin en eau (millimètres, mm)
- **Month:** Mois de l'année
- **Min Temp:** Température minimale (degrés Celsius, °C)
- **Max Temp:** Température maximale (degrés Celsius, °C)
- **Humidity:** Humidité relative (pourcentage %)
- **Wind:** Vitesse du vent (mètres par seconde, m/s)
- **Sun:** Ensoleillement (heures par jour, h/j)
- **Rad:** Radiation solaire (mégajoules par mètre carré par jour, MJ/m²/j)
- **Rain:** Précipitations (millimètres, mm)
- **Altitude:** Altitude (mètres, m)
- **Latitude:** Latitude (degrés)
- **Longitude:** Longitude (degrés)
- **Crop:** Type de culture (pomme de terre ou riz)
- **Soil:** Type de sol
- **City:** Ville.

Relations possibles entre ces es paramètres :

a) Water req et Crop:

Différentes cultures ont des besoins en eau différents. Le riz, par exemple, a tendance à nécessiter plus d'eau que la pomme de terre.

b) Min Temp, Max Temp et Crop:

La température minimale et maximale influence la croissance des cultures. Certaines cultures peuvent nécessiter des températures spécifiques pour un développement optimal.

c) Humidity et Water req:

L'humidité relative peut affecter les besoins en eau des cultures. Une humidité plus élevée pourrait réduire les besoins en irrigation.

d) Wind et Humidity:

La vitesse du vent peut influencer l'humidité relative, les vents forts ayant tendance à dessécher l'air et réduire l'humidité.

e) Sun et Crop:

La quantité d'ensoleillement reçue par une culture affecte sa croissance et ses besoins en eau. Plus de soleil peut augmenter l'évapotranspiration et les besoins en eau.

f) Rad et water req:

La radiation solaire affecte la photosynthèse et l'évapotranspiration, influençant ainsi les besoins en eau.

g) Rain et Water req:

Les précipitations contribuent directement aux besoins en eau des cultures. Des mois avec plus de pluie peuvent réduire les besoins d'irrigation.

h) Altitude, Latitude, Longitude et Crop:

Ces paramètres géographiques peuvent influencer le climat local, affectant ainsi les conditions de croissance des cultures.

i) Soil et Water req:

Le type de sol affecte la rétention d'eau et le drainage, influençant ainsi les besoins en irrigation des cultures.

Capture sur le fichier Arachides Riz :

CropDays	Soil Moisture	Soil Temperature	Temperature	Humidity	CropType	Irrigation(Y/N)
68	719.15826	19.407572	18.626808	60.326736	riz	Y
8	356.49728	20.688269	31.00492	65.082214	arachide	N
35	625.1673	20.813528	32.22308	58.391113	arachide	N
48	727.4161	19.588566	30.213121	67.49915	arachide	Y
81	119.61859	23.409452	17.422295	56.525246	arachide	N
10	696.9941	21.633026	28.16191	69.25261	arachide	Y
69	746.7571	18.368603	31.342724	59.13644	riz	N
84	663.88495	21.75674	18.107195	65.89839	arachide	N
46	676.50757	20.375065	32.982376	61.579727	arachide	Y

Capture sur le fichier Data_Plants :

water req	month	Min Temp	Max Temp	Humidity	Wind	Sun	Rad	Rain	altitude	latitude	longitude	crop	soil	city
83.990001	May	16.08	32.01	35.03	192.02	8.1	19.24	5.04	431.07	26.91	72.82	potato	red loamy	jaipur
0.07	March	26.05	33.05	54.01	168.08	5.02	20.6103	10.03	281.04	26.92	75.78	potato	red loamy	jaipur
39.07	May	21.02	32.03	62.01	168.09	9.02	22.08	10.0	14.05	19.17	72.82	rice	BLACK CLAY	kurnool
235.18	May	27.07	39.02	63.07	312.01	9.1	23.27	32.08	7.0	13.09	72.82	potato	red loamy	mumbai
0.07	May	26.05	33.05	54.01	168.08	5.02	20.6103	10.03	281.04	26.92	80.27	potato	red loamy	chennai
108.98	May	21.05	36.03	41.07	168.05	10.09	23.84	40.07	216.07	28.7598	72.82	rice	BLACK CLAY	kurnool
0.02	May	26.01	36.02	64.07	288.04	6.07	18.43	111.01	7.08	13.09	72.82	potato	red loamy	jaipur
116.19	May	25.01	36.02	70.06	192.0	9.29	23.35	51.08	9.02	22.73	72.82	rice	BLACK CLAY	kurnool
0.07	April	26.05	33.05	54.01	168.08	5.02	20.6103	10.03	281.04	26.92	88.36	potato	red loamy	kolkata

Base de données NoSQL

1. SQL (Structured Query Language):

SQL est un langage informatique normalisé servant à exploiter des bases de données **relationnelles**. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.

2. NoSQL (Not Only Structured Query Language):

Il désigne une catégorie de systèmes de gestion de base de données qui se différencient des systèmes de gestion de base de données **relationnelles traditionnels**. En effet, il utilise des modèles de données dits **non relationnels** qui offrent davantage de **flexibilité** : clés valeurs, graphes, colonnes et documents.

Les bases de données NoSQL sont généralement plus « **élastiques** » et **évolutives** que les bases de données relationnelles, et elles peuvent être distribuées sur plusieurs serveurs pour augmenter les performances et la tolérance aux pannes.

3. Pourquoi choisir NoSQL?

- a) **Structure de Données Flexible:** Les deux fichiers contiennent des enregistrements avec différents ensembles de champs et types de données. NoSQL (par exemple, MongoDB) permet une structure flexible où chaque document peut avoir un schéma différent, ce qui facilite l'intégration et la manipulation de données hétérogènes sans nécessiter de modifications structurelles complexes.
- b) **Scalabilité Horizontale:** Les bases de données NoSQL sont conçues pour une scalabilité horizontale facile. Si le volume de données augmente, il est possible d'ajouter plus de nœuds au cluster pour gérer la charge, ce qui est essentiel pour les applications d'agriculture à grande échelle qui recueillent des données en continu.
- c) **Performance:** Les bases de données NoSQL sont optimisées pour des requêtes en temps réel et peuvent offrir des performances supérieures pour certaines opérations, notamment lorsqu'il s'agit de grandes quantités de données non structurées ou semi-structurées.
- d) **Modélisation des Données Géospatiales:** Les bases de données NoSQL comme MongoDB supportent nativement les types de données géospatiales et les requêtes associées, ce qui est utile pour les données incluant des informations d'altitude, latitude, et longitude.

- e) **Adaptabilité aux données en temps réel:** Les environnements agricoles peuvent générer des données en temps réel nécessitant une ingérence rapide et efficace. Les systèmes NoSQL sont bien adaptés pour traiter et stocker des flux de données en continu.
- f) **Coût et Gestion:** Les bases de données NoSQL peuvent souvent être moins coûteuses à maintenir et à mettre à jour, surtout en termes de temps de développement, d'adaptation aux changements de schéma, et de scalabilité.

L'outil MongoDB

1) C'est quoi MongoDB?

MongoDB est une base de données **open-source** qui stocke les données sous forme de documents dans un format BSON (Binary JSON). BSON est une représentation binaire des documents JSON, permettant à MongoDB de combiner la flexibilité de JSON avec les performances binaires.

2) Format JSON:

JSON (JavaScript Object Notation) est un format textuel utilisé pour représenter des données structurées. Il est facile à lire et à écrire pour les humains et facile à parser et à générer pour les machines.

Exemple de document JSON :

```
{
  "_id": {
    "$oid": "6675c26e69f2f410cb3c1c47"
  },
  "CropDays": 81,
  "Soil Moisture": 119.61859,
  "Soil Temperature": 23.409452,
  "Temperature": 17.422295,
  "Humidity": 56.525246,
  "CropType": "arachide",
  "Irrigation(Y/N)": "N"
}
```

3) BSON:

BSON (Binary JSON) est une version binaire de JSON. Alors que JSON est textuel et lisible par l'homme, BSON est conçu pour être rapide et efficace en termes de stockage et de transmission.

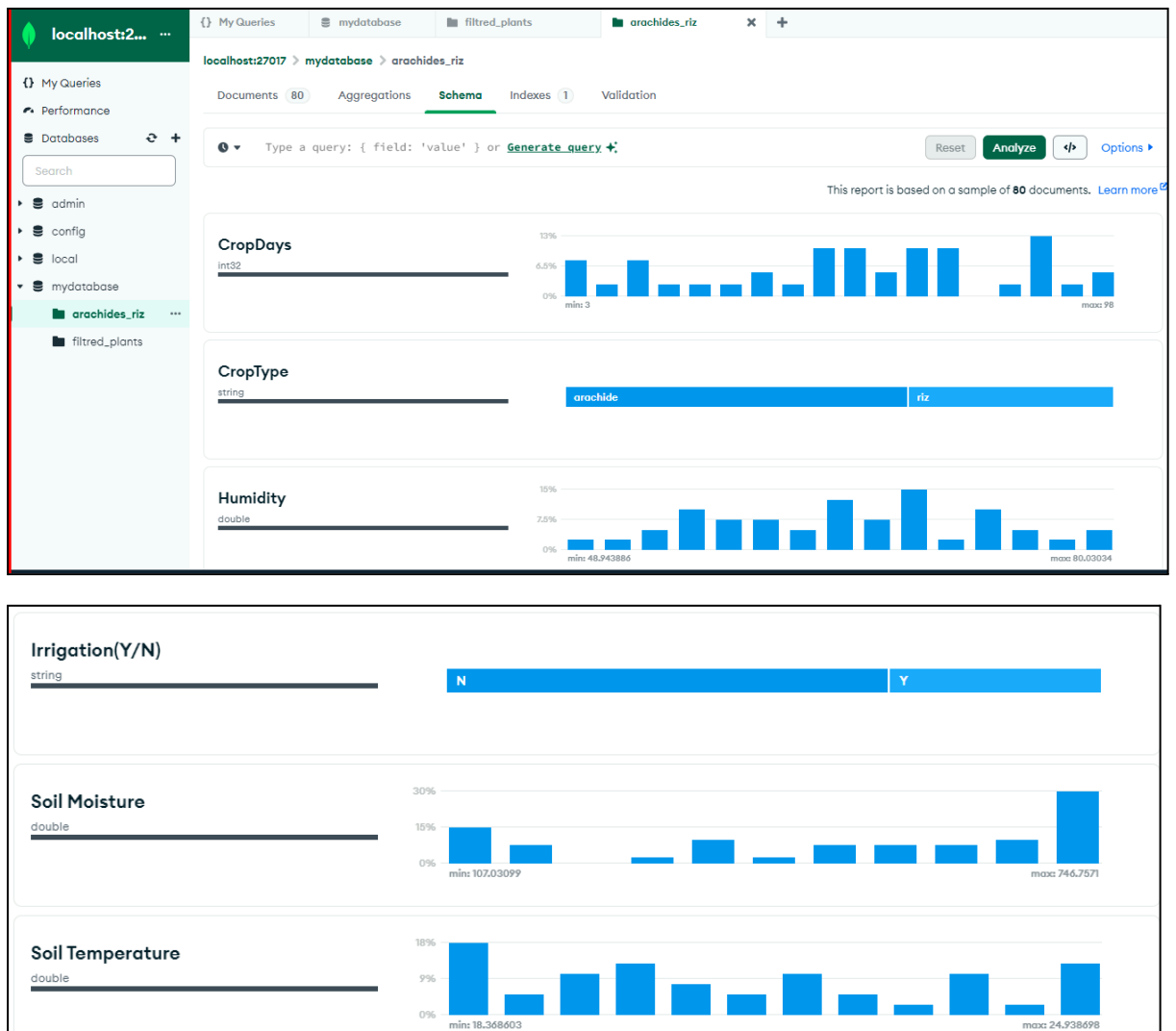
BSON encode les types de données, la longueur des chaînes, et d'autres métadonnées de manière binaire, ce qui permet des opérations de lecture et d'écriture plus rapides par rapport à JSON textuel.

4) Fonctionnalités offerte par MongoDB:

Les captures d'écran suivantes montrent l'analyse de schéma pour les collections `arachides_riz` et `filtered_plants` dans MongoDB Compass. Cette fonctionnalité permet de

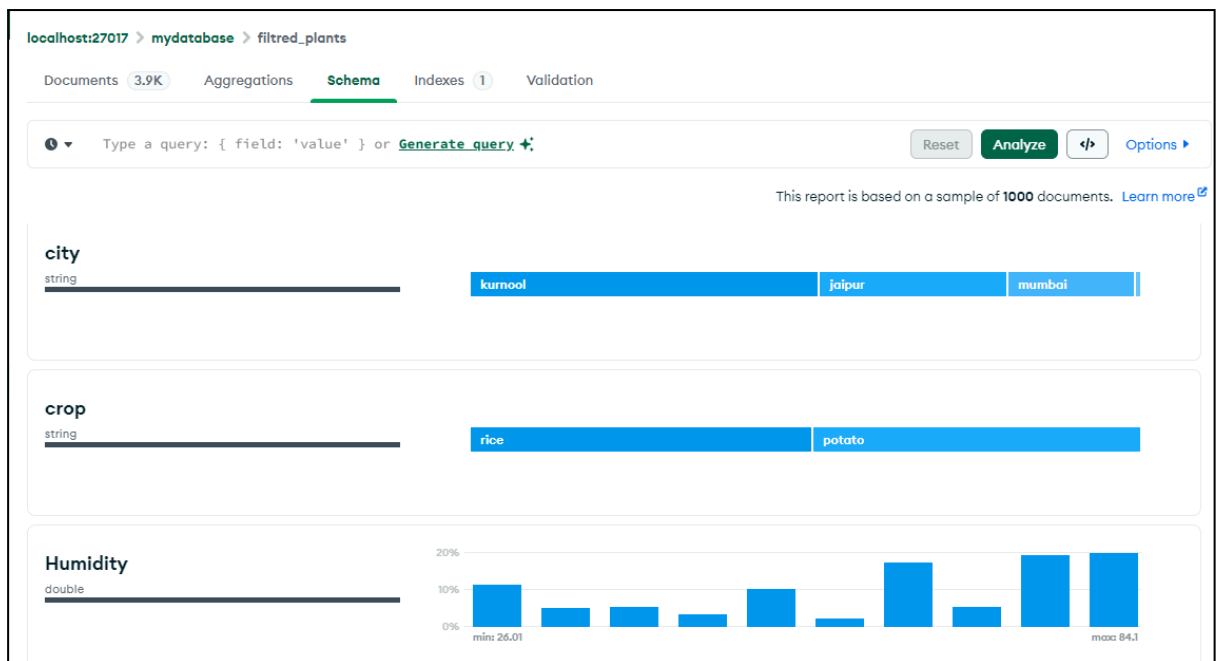
visualiser la structure des données et d'obtenir des informations statistiques sur les valeurs des champs.

a) Analyse de Schéma pour arachides_riz:



Cette capture d'écran montre les distributions des valeurs pour les champs CropDays, CropType, Humidity, Irrigation(Y/N), et Soil Moisture. Cela permet de comprendre rapidement la répartition des données et d'identifier des anomalies potentielles.

b) Analyse de Schéma pour filtred_plants:



Cette capture d'écran montre les distributions des valeurs pour les champs **city**, **crop**, et **Humidity**. MongoDB Compass permet de visualiser les données de manière interactive, facilitant ainsi l'analyse et l'optimisation des requêtes.

c) **Vue d'ensemble des Collections:**

The screenshot shows the 'Collections' view in MongoDB Compass. It lists two collections: 'arachides_riz' and 'filtred_plants'. For each collection, it provides details on storage size, number of documents, average document size, number of indexes, and total index size.

Collection Name	Storage size	Documents	Avg. document size	Indexes	Total index size
arachides_riz	20.48 kB	80	168.00 B	1	36.86 kB
filtred_plants	323.58 kB	3.9 K	273.00 B	1	81.92 kB

Cette capture d'écran montre la vue d'ensemble des collections dans MongoDB Compass. Elle fournit des détails sur la taille de stockage, le nombre de documents, la taille moyenne des documents, le nombre d'index et la taille totale des index pour chaque collection (arachides_riz et filtred_plants). Cela offre une vue globale de la base de données et de sa structure, facilitant la gestion et l'optimisation des données.

Réalisation des requêtes en MongoDB

1) Configuration de l'Environnement :

a. Installation de MongoDB

- Téléchargement et installation de MongoDB (version 7.0.11).

b. Environnement Python :

- `pip install pandas pymongo openpyxl`

Pour ce projet, nous avons utilisé plusieurs de ces bibliothèques Python pour manipuler et interagir avec nos données :

- **pandas**: Bibliothèque utilisée pour **lire et manipuler** les fichiers de données CSV et Excel. Elle nous a permis de **convertir** les fichiers en structures de données tabulaires (DataFrames) faciles à manipuler.
- **pymongo**: Bibliothèque utilisée pour **se connecter à MongoDB** et interagir avec la base de données. Nous avons utilisé Pymongo pour **insérer les données** dans MongoDB et **exécuter des requêtes** pour extraire les informations nécessaires.
- **openpyxl**: Bibliothèque utilisée par pandas pour lire les fichiers Excel. Elle a permis à pandas de charger les données des fichiers Excel directement dans des DataFrames.

2) Chargement des Données dans MongoDB :

a. Lecture et de Conversion des Données :

- La fonction `pandas.read_csv()` nous a permis de lire les fichiers CSV et de les convertir en DataFrame.
- La fonction `to_json()` nous a permis de convertir les DataFrames en JSON pour les rendre compatibles avec MongoDB.

```
import pandas as pd
import json

# Chargement des fichiers CSV
arachides_riz_df = pd.read_csv('data/arachides_riz_generated.csv')
filtred_plants_df = pd.read_csv('data/filtred_plants.csv')

# Conversion en JSON
arachides_riz_json = arachides_riz_df.to_json(orient='records')
```

```
filtred_plants_json = filtred_plants_df.to_json(orient='records')

# Sauvegarde des fichiers JSON
with open('data/arachides_riz_generated.json', 'w') as file:
    file.write(arachides_riz_json)

with open('data/filtred_plants.json', 'w') as file:
    file.write(filtred_plants_json)
```

Exemples de données JSON

Voici des exemples de données JSON après conversion, qui montrent la structure des documents insérés dans MongoDB.

- Données sur les arachides et le riz :

```
{
  "_id": {
    "$oid": "6675c26e69f2f410cb3c1c47"
  },
  "CropDays": 81,
  "Soil Moisture": 119.61859,
  "Soil Temperature": 23.409452,
  "Temperature": 17.422295,
  "Humidity": 56.525246,
  "CropType": "arachide",
  "Irrigation(Y/N)": "N"
}
```

- Données sur les plantes :

```
{
  "_id": {
    "$oid": "6675c2d1248f453854c34e7b"
  },
  "water req": 235.18,
  "month": "May",
  "Min Temp": 27.07,
  "Max Temp": 39.02,
  "Humidity": 63.07,
  "Wind": 312.01,
  "Sun": 9.1,
  "Rad": 23.27,
  "Rain": 32.08,
  "altitude": 7,
  "latitude": 13.09,
  "longitude": 72.82,
  "crop": "potato",
  "soil": "red loamy",
  "city": "mumbai"
}
```

b. Insertion des données dans MongoDB à l'aide de pymongo :

Nous avons utilisé PyMongo pour effectuer une lecture sur les fichiers JSON convertis depuis des DataFrames pandas et ensuite nous avons inséré les données dans les collections MongoDB avec la fonction `insert_many()`.

```
from pymongo import MongoClient
import json

# Connexion à MongoDB
client = MongoClient('mongodb://localhost:27017/')
db = client['mydatabase']

# Lecture des fichiers JSON
with open('data/arachides_riz_generated.json', 'r') as file:
    arachides_riz_data = json.load(file)

with open('data/filtred_plants.json', 'r') as file:
    filtred_plants_data = json.load(file)

# Insertion des données dans les collections MongoDB
db.arachides_riz.insert_many(arachides_riz_data)
db.filtred_plants.insert_many(filtred_plants_data)

print("Données insérées avec succès.")
```

3) Interrogation des Données dans MongoDB

a. Sur le premier fichier (Arachides et Riz):

N° Req	Requête
01	<p>Cette requête calcule les moyennes de la température du sol (Soil Temperature) et de l'humidité du sol (Soil Moisture) en fonction du type de culture et de l'irrigation.</p> <pre>db.arachides_riz.aggregate([{ \$group: { _id: { cropType: "\$CropType", irrigation: "\$Irrigation(Y/N)" }, averageSoilTemp: { \$avg: "\$Soil Temperature" }, averageSoilMoisture: { \$avg: "\$Soil Moisture" } } }, {</pre>

	<pre> \$project: { _id: 0, cropType: "\$_id.cropType", irrigation: "\$_id.irrigation", averageSoilTemp: 1, averageSoilMoisture: 1 } }]) </pre>
--	---

Résultat ->

Crop Type	Irrigation	Average Soil Temperature	Average Soil Moisture
arachide	Y	20.1603	683.862
riz	Y	20.1052	607.816
arachide	N	21.7505	392.996
riz	N	21.7472	436.009

02

Requête pour trouver les enregistrements où les conditions environnementales sont extrêmes (Température > 30 ou Humidité < 60) et nécessitent une irrigation.

```

db.arachides_riz.find({
  "Irrigation(Y/N)": "Y",
  $or: [
    { Temperature: { $gt: 30 } },
    { Humidity: { $lt: 60 } }
  ]
})

```

Résultat ->

Crop Type	Irrigation	Temperature	Humidity	Soil Moisture	Soil Temperature	ID
arachide	Y	30.2131	67.4992	727.416	19.5886	6675c26e69f2f410cb3c1c48
arachide	Y	32.9824	61.5797	676.508	20.3751	6675c26e69f2f410cb3c1c51
arachide	Y	29.1628	52.197	522.325	19.4926	6675c26e69f2f410cb3c1c57
arachide	Y	30.6242	58.0514	728.448	21.0173	6675c26e69f2f410cb3c1c5b
arachide	Y	27.898	54.4047	715.65	18.5948	6675c26e69f2f410cb3c1c5c
arachide	Y	30.2131	67.4992	727.416	19.5886	6675c9dfcd7507a40dd780
arachide	Y	32.9824	61.5797	676.508	20.3751	6675c9dfcd7507a40dd781
arachide	Y	30.6242	58.0514	728.448	21.0173	6675c9dfcd7507a40dd78f
arachide	Y	29.1628	52.197	522.325	19.4926	6675c9dfcd7507a40dd793
arachide	Y	27.898	54.4047	715.65	18.5948	6675c9dfcd7507a40dd796

03

Requête pour générer un rapport détaillé des cultures avec le **maximum, minimum et moyenne des jours de culture** pour chaque **type de culture**.

```
db.arachides_riz.aggregate([
  {
    $group: {
      _id: "$CropType",
      maxCropDays: { $max: "$CropDays" },
      minCropDays: { $min: "$CropDays" },
      avgCropDays: { $avg: "$CropDays" }
    }
  }
])
```

Résultat ->

Crop Type	Max Crop Days	Min Crop Days	Avg Crop Days
arachide	98	8	53.4
riz	91	3	48.6

04

Requête pour trouver les enregistrements où l'irrigation est nécessaire et où la température du sol et l'humidité sont **supérieures à la moyenne** pour chaque type de culture.

```
db.arachides_riz.aggregate([
  {
    $group: {
      _id: "$CropType",
      avgSoilTemperature: { $avg: "$Soil Temperature" },
      avgHumidity: { $avg: "$Humidity" }
    }
  },
  {
    $lookup: {
      from: "arachides_riz",
      let: { type: "$_id", avgTemp: "$avgSoilTemperature",
        avgHum: "$avgHumidity" },
      pipeline: [
        {
          $match: {
            $expr: {
              $and: [
                { $eq: ["$CropType", "$$type"] },
                { $gt: ["$Soil Temperature", "$$avgTemp"] },
                { $gt: ["$Humidity", "$$avgHum"] },
                { $eq: ["$Irrigation(Y/N)", "Y"] }
              ]
            }
          }
        }
      ]
    }
  }
])
```

	<pre>] } } }, as: "records" } }, { \$unwind: "\$records" }, { \$replaceRoot: { newRoot: "\$records" } }]) </pre>
--	---

Résultat ->

Crop Type	Irrigation	Soil Temperature	Humidity	ID
arachide	Y	21.633	69.2526	6675c26e69f2f410cb3c1c56
arachide	Y	21.633	69.2526	6675c9dfcd7507a40dd78a
riz	Y	23.6311	71.9798	6675c26e69f2f410cb3c1c4f
riz	Y	23.6311	71.9798	6675c9dfcd7507a40dd784

b. Sur le deuxième Fichier (Data Plants):

N° Req	Requête
01	<p>Calculer la moyenne des températures minimales et maximales pour chaque type de culture dans chaque ville.</p> <pre> db.filtred_plants.aggregate([{ \$group: { _id: { crop: "\$crop", city: "\$city" }, avgMinTemp: { \$avg: "\$Min Temp" }, avgMaxTemp: { \$avg: "\$Max Temp" } } }, { \$sort: { _id: 1 } }]) </pre>
Résultat ->	

Crop Type	City	Avg Min Temp	Avg Max Temp
potato	chennai	26.05	33.05
potato	delhi	26.05	33.05
potato	jaipur	25.0959	35.3954
potato	kolkata	26.05	33.05
potato	kurnool	26.05	33.05
potato	mumbai	24.829	35.236
rice	chennai	26.05	33.05
rice	delhi	26.05	33.05
rice	jaipur	26.05	33.05
rice	kolkata	26.05	33.05
rice	kurnool	25.0223	35.2827
rice	mumbai	26.05	33.05

o2

Trouver les enregistrements avec des besoins en eau supérieurs à la moyenne pour chaque type de culture.

```
db.filtered_plants.aggregate([
  {
    $group: {
      _id: "$crop",
      moyenneBesoinEau: { $avg: "$water req" }
    }
  },
  {
    $addFields: {
      moyenneBesoinEau: {
        $arrayElemAt: [
          {
            $filter: {
              input: moyennes_besoins_eau,
              as: "item",
              cond: { $eq: ["$$item._id", "$crop"] }
            }
          },
          0
        ]
      }
    }
  }
])
```

	<pre> }, { \$match: { \$expr: { \$gt: ["\$water req", "\$moyenneBesoinEau.moyenneBesoinEau"] } } }, { \$limit: 5 }]) </pre>
--	---

Résultat ->

Crop Type	Water Requirement	Average Water Requirement	City	Soil Type
potato	235.18	114.202	mumbai	red loamy
rice	116.19	113.608	kurnool	BLACK CLAY
potato	221.72	114.202	mumbai	red loamy
rice	141.58	113.608	kurnool	BLACK CLAY
rice	185.71	113.608	kurnool	BLACK CLAY
potato	236.65	114.202	jaipur	red loamy

03

Trouver les cultures avec des conditions de vent et de radiation solaires élevées dans des villes avec une altitude inférieure à 100 mètres. La requête sélectionne les six premiers enregistrements répondant à ces critères.

```

db.filtred_plants.find({
  "altitude": { "$lt": 100 },
  "Wind": { "$gt": 200 },
  "Rad": { "$gt": 20 }
}).limit(6)

```

Résultat ->

Crop Type	City	Altitude	Wind Speed	Solar Radiation
potato	mumbai	7	312.01	23.27
potato	jaipur	7.09	312.05	23.2
potato	mumbai	7.05	216.08	23.51
rice	kurnool	7.04	312.08	23.25
rice	kurnool	7.02	264.01	24.59
rice	kurnool	14.09	240.07	25

04

Générer un rapport des cultures par mois avec la somme des besoins en eau, groupé par type de sol.

```
db.filtred_plants.aggregate([
  {
    $group: {
      _id: {
        month: "$month",
        soil: "$soil"
      },
      totalWaterReq: { $sum: "$water req" }
    }
  },
  {
    $sort: { "_id.month": 1 }
  }
])
```

Résultat ->

Month	Soil Type	Total Water Requirement
April	red loamy	0.84
August	BLACK CLAY	0.84
July	BLACK CLAY	0.84
July	red loamy	0.84
June	red loamy	0.84
June	BLACK CLAY	0.84
March	red loamy	0.84
May	BLACK CLAY	223805
May	red loamy	224975
September	BLACK CLAY	0.84

05

Trouver les cultures qui nécessitent le plus d'eau pendant le mois de mai.

	<pre> db.filtred_plants.aggregate([{ \$match: { month: "May" } }, { \$group: { _id: "\$crop", averageWaterReq: { \$avg: "\$water req" } } }, { \$sort: { averageWaterReq: -1 } }]) </pre>
--	---

Résultat ->

Crop Type	Average Water Requirement
potato	117.053
rice	116.444

o6	<p>Trouver les cultures ayant des températures minimales et maximales les plus élevées et les plus basses pour chaque mois.</p> <pre> db.filtred_plants.aggregate([{ \$group: { _id: { month: "\$month", crop: "\$crop" }, maxMaxTemp: { \$max: "\$Max Temp" }, minMaxTemp: { \$min: "\$Max Temp" }, maxMinTemp: { \$max: "\$Min Temp" }, minMinTemp: { \$min: "\$Min Temp" } } }, { \$sort: { "_id.month": 1 } }]) </pre>
----	--

Résultat ->

Month	Crop Type	Max Max Temp	Min Max Temp	Max Min Temp	Min Min Temp
April	potato	33.05	33.05	26.05	26.05
August	rice	33.05	33.05	26.05	26.05
July	rice	33.05	33.05	26.05	26.05
July	potato	33.05	33.05	26.05	26.05
June	rice	33.05	33.05	26.05	26.05
June	potato	33.05	33.05	26.05	26.05
March	potato	33.05	33.05	26.05	26.05
May	rice	41.1	29	28.1	15
May	potato	41.1	29.01	28.1	15
September	rice	33.05	33.05	26.05	26.05

c. Requête combinées sur les deux fichiers:

N° Req	Requête
01	<p>Cette requête retourne les moyennes des paramètres de culture pour le riz en mai, groupées par type de sol et par ville. Les paramètres incluent :</p> <ul style="list-style-type: none"> • averageWaterReq : La moyenne des besoins en eau. • averageTemp : La moyenne de la température. • averageHumidity : La moyenne de l'humidité du sol (en pourcentage). • averageMinTemp : La moyenne de la température minimale. • averageMaxTemp : La moyenne de la température maximale. • averageEnvHumidity : La moyenne de l'humidité environnementale. • averageSoilMoisture : La moyenne de l'humidité du sol. • averageSoilTemp : La moyenne de la température du sol. <pre> db.arachides_riz.aggregate([{ \$addFields: { normalized_crop_type: { \$cond: [{ \$eq: ["\$CropType", "riz"] }, "rice", "\$CropType"] } } }, { \$match: { normalized_crop_type: "rice" } }]) </pre>


```

    }
  },
  {
    $lookup: {
      from: "filtred_plants",
      localField: "normalized_crop_type",
      foreignField: "crop",
      as: "matched_env"
    }
  },
  {
    $unwind: "$matched_env"
  },
  {
    $match: {
      "matched_env.month": "May"
    }
  },
  {
    $group: {
      _id: {
        soil: "$matched_env.soil",
        city: "$matched_env.city"
      },
      averageWaterReq: { $avg: "$matched_env.water req" },
      averageTemp: { $avg: "$Temperature" },
      averageHumidity: { $avg: "$matched_env.Humidity" }, #
Humidité du sol
      averageMinTemp: { $avg: "$matched_env.Min Temp" },
      averageMaxTemp: { $avg: "$matched_env.Max Temp" },
      averageEnvHumidity: { $avg: "$matched_env.Humidity" },
      averageSoilMoisture: { $avg: "$Soil Moisture" },
      averageSoilTemp: { $avg: "$Soil Temperature" }
    }
  },
  {
    $project: {
      _id: 0,
      soil: "$_id.soil",
      city: "$_id.city",
      averageWaterReq: 1,
      averageTemp: 1,
      averageHumidity: 1, # Humidité du sol
      averageMinTemp: 1,

```

	<pre> averageMaxTemp: 1, averageEnvHumidity: 1, averageSoilMoisture: 1, averageSoilTemp: 1 } }]) </pre>
--	---

Résultat ->

Soil Type	City	Avg Water Req	Avg Temp	Avg Soil Humidity	Avg Min Temp	Avg Max Temp	Avg Env Humidity	Avg Soil Moisture	Avg Soil Temp
BLACK CLAY	delhi	0.07	24.5183	54.01	26.05	33.05	54.01	504.732	21.0904
BLACK CLAY	kurnool	117.053	24.5183	60.0471	25.018	35.2921	60.0471	504.732	21.0904
BLACK CLAY	kolkata	0.07	24.5183	54.01	26.05	33.05	54.01	504.732	21.0904
BLACK CLAY	chennai	0.07	24.5183	54.01	26.05	33.05	54.01	504.732	21.0904
BLACK CLAY	jaipur	0.07	24.5183	54.01	26.05	33.05	54.01	504.732	21.0904
BLACK CLAY	mumbai	0.07	24.5183	54.01	26.05	33.05	54.01	504.732	21.0904

Conclusion

Notre étude a permis de démontrer l'efficacité de MongoDB en tant que base de données NoSQL adaptée à la gestion et à l'analyse de données agronomiques complexes. En choisissant MongoDB, nous avons pu bénéficier de sa flexibilité, de sa capacité à gérer des structures de données variées et de sa scalabilité horizontale, toutes essentielles pour les applications agricoles à grande échelle.

Nous avons procédé de la manière suivante :

- 1) **Détermination de la structure de données:** Les fichiers CSV ont été analysés et convertis en format JSON pour être compatibles avec MongoDB.
- 2) **Choix de l'outil:** MongoDB a été choisi pour ses avantages en termes de flexibilité, de performance et de modélisation des données géospatiales.
- 3) **Création de la base de données:** Les données ont été importées dans MongoDB et organisées en collections.
- 4) **Requêtes stratégiques:** Des requêtes spécifiques ont été exécutées pour analyser les besoins en eau des cultures, l'impact de l'irrigation et d'autres paramètres climatiques.

Les captures d'écran incluses dans ce rapport illustrent la visualisation des schémas de données dans MongoDB Compass, ce qui a facilité l'analyse et l'identification des anomalies potentielles. Grâce à cette approche, nous avons pu obtenir des insights précieux pour l'optimisation des pratiques agricoles, démontrant ainsi la valeur ajoutée de MongoDB dans ce contexte.

Bibliographie

Cours BDDA SIQ, Mme Amrouche Karima, ESI