

MASTÈRE SPÉCIALISÉ EXPERT EN SCIENCE DES DONNÉES

Informatique 1

Pâquarse Delvich Van Mahouvi

Novembre 2023

1 Partie 1 : Réponse aux questions sous MongoDB

Dans le but d'apprendre le fonctionnement de différentes bases de données NoSQL, nous avons opté pour la modélisation Cassandra et MongoDB pour la première partie. Le but étant d'apprendre un autre type de base de données autre que MongoDB, la deuxième partie étant à faire uniquement sous MongoDB. Pour les modélisations Cassandra, un container Cassandra a été déployer sur Docker. Ainsi, à la suite de chaque réponse pour la partie 1, la requête MongoDB permettant d'avoir le résultat sera également proposé.

```
[ ]: docker run -d -v C:/Users/Paqua/OneDrive/Documents/Mastere_ESD/Info1/Projet/:/
    ↪Data --name delvich-cassandra cassandra
docker exec -it delvich-cassandra bash
WITH replication = {'class':'SimpleStrategy', 'replication_factor': 1};
use exutoire;
```

1. Quel est le nombre d'exutoires pour les déchets Dématérialisation ?

```
[ ]: CREATE TABLE nombre_exutoire(
    label TEXT,
    id_exutoire INT,
    type TEXT,
    description TEXT,
    id_dechet INT,
    nom_dechet TEXT,
    famille_dechet TEXT,
    PRIMARY KEY (label, id_dechet)
);
COPY nombre_exutoire (id_exutoire, label, type, description, id_dechet,
    ↪nom_dechet, famille_dechet) FROM '/Data/relation_dechets_exutoires.csv' WITH
    ↪DELIMITER = ';' AND HEADER = TRUE;
SELECT label, COUNT(*) as "Nombre d'exutoire"
FROM nombre_exutoire
WHERE label = 'Dématérialisation';
```

label	Nombre d'exutoire
Dématérialisation	14

Sous MongoDB, le même résultat peut être obtenu en exécutant :

```
[ ]: db.Exutoire.countDocuments({
    "label": "Dématérialisation"
});
```

2. Quel est le type d'exutoire pour les déchets de Peinture à l'eau ?

```
[ ]: CREATE TABLE type_extutoire_par_dechet(
    label TEXT,
    id_exutoire INT,
    type TEXT,
    description TEXT,
    id_dechet INT,
    nom_dechet TEXT,
    famille_dechet TEXT,
    PRIMARY KEY(nom_dechet, type)
);

SELECT type, nom_dechet
FROM type_extutoire_par_dechet
WHERE nom_dechet = 'Peinture à l'eau';
```

type	nom_dechet
Conseil	Peinture à l'eau
Exutoire	Peinture à l'eau

```
[ ]: db.Exutoire.find({
    "nom_dechet": "Peinture à l'eau"
});
```

3. Quels sont les déchets associés à l'exutoire id 3

```
[ ]: CREATE TABLE nom_dechet_par_id_exutoire (
    label TEXT,
    id_exutoire INT,
    type TEXT,
    description TEXT,
    id_dechet INT,
    nom_dechet TEXT,
    famille_dechet TEXT,
    PRIMARY KEY (id_exutoire, id_dechet)
);

SELECT id_exutoire, nom_dechet
FROM nom_dechet_par_id_exutoire
WHERE id_exutoire = 3;
```

id_exutoire	nom_dechet
3	Bocal en verre
3	Bouteille en verre
3	Flacon en verre
3	Pot de yaourt en verre
3	Pot en verre

```
[ ]: db.Exutoire.find({
  "id_exutoire": 3
}, {
  "_id": 0,
  "id_dechet": 1,
  "nom_dechet": 1
}).toArray();
```

4. Quels sont les déchets qui ont comme famille Verre ?

```
[ ]: CREATE TABLE dechets_par_famille_de_dechet(
  label TEXT,
  id_exutoire INT,
  type TEXT,
  description TEXT,
  id_dechet INT,
  nom_dechet TEXT,
  famille_dechet TEXT,
  PRIMARY KEY (famille_dechet, nom_dechet)
);

SELECT nom_dechet, famille_dechet
FROM dechets_par_famille_de_dechet
WHERE famille_dechet = 'Verre';
```

nom_dechet	famille_dechet
Bocal en verre	Verre
Bouteille en verre	Verre
Flacon en verre	Verre
Pot de yaourt en verre	Verre
Pot en verre	Verre

```
[ ]: use('ExutoireStore')
db.Exutoire.find({
  "famille_dechet": "Verre"
}, {
  "_id": 0,
  "famille_dechet": 1,
  "id_dechet": 1,
  "nom_dechet": 1,
}).toArray();
```

5. Quelle est la description de l'exutoire 6

```
[ ]: SELECT id_exutoire, description
FROM nom_dechet_par_id_exutoire
WHERE id_exutoire = 6
LIMIT 1;
```

id_exutoire	description
6	Ce déchet peut être composté. Il deviendra alors une ressource !!

```
[ ]: db.Exutoire.findOne({
  "id_exutoire": 6
}, {
  "_id": 0,
  "description": 1
});
```

6. Combien de type de déchets sont concernés par les exutoires de type Conseil

```
[ ]: CREATE TABLE famille_dechet_par_exutoire(
  label TEXT,
  id_exutoire INT,
  type TEXT,
  description TEXT,
  id_dechet INT,
  nom_dechet TEXT,
  famille_dechet TEXT,
  PRIMARY KEY (type, famille_dechet)
);

SELECT type, COUNT(famille_dechet) as nombre_famille_dechet
FROM famille_dechet_par_exutoire
WHERE type = 'Conseil';
```

type	nombre_famille_dechet
Conseil	22

```
[ ]: db.Exutoire.distinct("nom_dechet", {
  "type": "Conseil"
});
```

7. Quel est le type d'exutoire qui contient le plus de déchets différents ?

```
[ ]: CREATE TABLE nom_dechet_par_type_exutoire(
  label TEXT,
  id_exutoire INT,
  type TEXT,
  description TEXT,
  id_dechet INT,
  nom_dechet TEXT,
  famille_dechet TEXT,
  PRIMARY KEY (type, id_dechet, famille_dechet)
)WITH CLUSTERING ORDER BY (id_dechet DESC);
```

```
SELECT type, COUNT (id_dechet) AS "Nombre de type de déchet"
FROM nom_dechet_par_type_exutoire
GROUP BY type;
```

type	Nombre de type de déchet
Exutoire	1447
Conseil	1144

```
[ ]: db.Exutoire.aggregate([
  {
    $group: {
      _id: "$type",
      count: { $addToSet: "$nom_dechet" }
    }
  },
  {
    $project: {
      _id: 1,
      count: { $size: "$count" }
    }
  },
  {
    $sort: { count: -1 }
  },
  {
    $limit: 1
  }
]);
```

8. Quel est le type d'exutoire le plus courant ?

```
[ ]: SELECT type, COUNT(famille_dechet) as nombre_famille_dechet
FROM famille_dechet_par_exutoire;
```

type	nombre_famille_dechet
Exutoire	45

```
[ ]: db.Exutoire.aggregate([
  {
    $group: {
      _id: "$type",
      count: { $sum: 1 }
    }
  },
  {
    $sort: { count: -1 }
  },
  {
    $limit: 1
  }
]);
```

```

    $sort: { count: -1 }
  },
  {
    $limit: 1
  }
}
]);

```

9. Quel est le type d'exutoire le plus courant par famille de déchet ?

```

[ ]: CREATE TABLE famille_par_type_exutoire(
  label TEXT,
  id_exutoire INT,
  type TEXT,
  description TEXT,
  id_dechet INT,
  nom_dechet TEXT,
  famille_dechet TEXT,
  PRIMARY KEY ((famille_dechet, type), id_dechet)
);

COPY famille_par_type_exutoire(id_exutoire, label, type, description, id_dechet,
  ↳nom_dechet, famille_dechet) FROM '/Data/relationdechets_exutoires.csv' WITH
  ↳DELIMITER = ';' AND HEADER = TRUE;

SELECT famille_dechet, type, COUNT (id_exutoire) as Nombre
FROM famille_par_type_exutoire
GROUP BY famille_dechet, type;

```

famille_dechet	type	nombre
Sport	Exutoire	36
Bijoux	Exutoire	8
Alimentation	Conseil	153
Divers	Conseil	2
Textiles d'habillement, linge de maison et chaussures	Exutoire	108
Culture loisirs	Conseil	9
Bois	Conseil	7
Outils manuel	Conseil	33
Papier carton	Conseil	28
Produits chimiques	Conseil	19
Emballages en métal	Exutoire	22
Instrument de musique	Exutoire	27
Divers	Exutoire	14
Transport	Conseil	21
Equipement de la maison	Conseil	6
Médical	Exutoire	15
Electrique et électronique	Conseil	262
Emballages en métal	Conseil	8
Outils manuel	Exutoire	33
Vie domestique	Exutoire	327
Papier carton	Exutoire	67
Objets en Plastique	Conseil	17
Electrique et électronique	Exutoire	275
Bois	Exutoire	12
Emballages en plastique	Conseil	15
Produits chimiques	Exutoire	68
Transport	Exutoire	27
Matériaux	Conseil	113
Emballages en plastique	Exutoire	39
Textiles d'habillement, linge de maison et chaussures	Conseil	104
Verre	Exutoire	5
Végétaux	Conseil	29
Végétaux	Exutoire	36
Matériaux	Exutoire	122
Verre	Conseil	3
Alimentation	Exutoire	164
Culture loisirs	Exutoire	10
Vaisselle	Conseil	11
Objets en Plastique	Exutoire	25
Instrument de musique	Conseil	27
Equipement de la maison	Exutoire	6
Vaisselle	Exutoire	13
Vie domestique	Conseil	238
Sport	Conseil	36
Bijoux	Conseil	9

```
[ ]: db.Exutoire.aggregate([
  {
    $group: {
      _id: { type: "$type", famille_dechet: "$famille_dechet" },
      count: { $sum: 1 }
    }
  },
  {
    $sort: { "_id.famille_dechet": 1, count: -1 }
  },
  {
    $group: {
      _id: "$_id.famille_dechet",
      type: { $first: "$_id.type" },
      count: { $first: "$count" }
    }
  },
  {
    $sort: { count: -1 }
  }
]);
```

10. Quels sont les labels d'exutoires associés à chaque famille de déchets ?

```
[ ]: CREATE TABLE label_par_famille(
  label TEXT,
  id_exutoire INT,
  type TEXT,
  description TEXT,
  id_dechet INT,
  nom_dechet TEXT,
  famille_dechet TEXT,
  PRIMARY KEY (famille_dechet, label)
)WITH CLUSTERING ORDER BY (label ASC);

COPY label_par_famille (id_exutoire, label, type, description, id_dechet,
→nom_dechet, famille_dechet) FROM '/Data/relation_dechets_exutoires.csv' WITH
→DELIMITER = ';' AND HEADER = TRUE;

SELECT famille_dechet, label
FROM label_par_famille
WHERE famille_dechet = 'Culture loisirs';
```

famille_dechet	label
Culture loisirs	Don / Entraide / Revente
Culture loisirs	Déchetterie
Culture loisirs	Dématérialisation
Culture loisirs	Poubelle noire
Culture loisirs	Poubelle verte / jaune

L'affichage étant très grand pour le document, on a opter pour afficher uniquement les déchets de la famille 'Culture loisirs'. Cet élément pourra être retiré pour afficher tous les labels d'exutoire par famille de déchets. Cet élément est également fourni par la requête MongoDB ci-dessous.

```
[ ]: use('ExutoireStore')
db.Exutoire.aggregate([
  {
    $group: {
      _id: { famille_dechet: "$famille_dechet" },
      labels: { $addToSet: "$label" }
    }
  }
]);
```

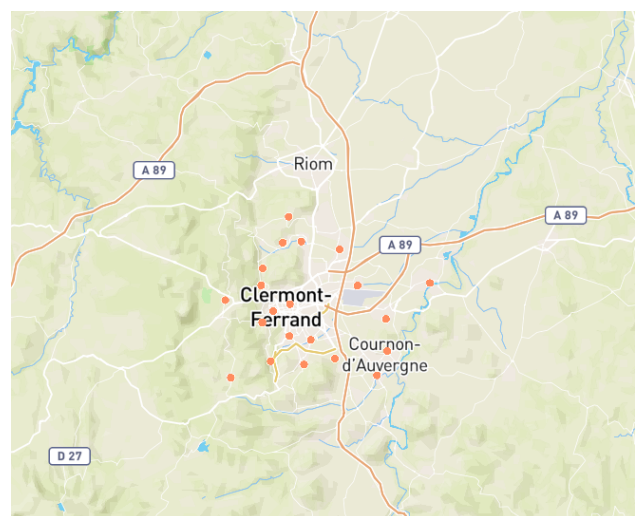
11- Indiquer une clé qui permettrait de s'assurer que les données soient réparties de manière uniforme sur un cluster distribué.

Il faudra choisir une clé de sharding de cardinalité élevée, c'est-à-dire un grand nombre de valeurs uniques. Cela contribue à répartir uniformément les données. Dans notre cas, l'identifiant des déchets semble être un excellent choix.

12- La clé qui permettrait de ne traiter les requêtes ne concernant que les déchets sur un seul nœud est la clé `id_exutoire`.

2 Partie 2 : Réponse aux questions sous MongoDB

Le fichier *atmo-indice.geojson* a été importé sous Mongo grâce à MongoCompass et une connexion sur VSCode. Dans le but de rester fidèle le plus possible aux syntaxes MongoDB, nous avons donc opté pour des captures d'écran, en lieu et place d'un fichier jupyter permettant d'utiliser pymongo. Ainsi, des captures d'écran seront faites pour la suite. En représentant les données, nous pouvons observer qu'il s'agit de différentes communes de la zone de Clermont Auvergne Métropole.



1- Quels sont les communes qui ont eu un indice de qualité de l'air Moyen ?

```
[ ]: db.Atmosphere.distinct("properties.lib_zone", {
    "properties.lib_qual": "Moyen",
    "properties.type_zone": "commune"
})
```

```
[5]: %%js
[
    "Aubière",
    "Aulnat",
    "Beaumont",
    "Blanzat",
    "Ceyrat",
    "Chamalières",
    "Châteaugay",
    "Clermont-Ferrand",
    "Cournon-d'Auvergne",
    "Cébazat",
    "Durtol",
    "Gerzat",
    "Le Cendre",
    "Lempdes",
    "Nohanent",
    "Orcines",
    "Pont-du-Château",
    "Pérignat-lès-Sarliève",
    "Romagnat",
    "Royat",
    "Saint-Genès-Champanelle"
]
```

<IPython.core.display.Javascript object>

2- Quelle est le code de qualité de l'air maximal sur la commune de Saint-Genès-Champanelle

```
[ ]: db.Atmosphere.find({"properties.lib_zone": "Saint-Genès-Champanelle"},
    {"properties.lib_zone":1, "properties.code_qual":1}).sort({ "properties.
    ↪code_qual": -1 }).limit(1)
```

```
[4]: %%js
[
    {
        "_id": {
            "$oid": "655809d77a1790cd7bce62f8"
        },
        "properties": {
            "code_qual": 3,
            "lib_zone": "Saint-Genès-Champanelle"
        }
    }
]
```

```
}  
]
```

<IPython.core.display.Javascript object>

3- Quelles sont les différentes communes observées ?

```
[ ]: db.Atmosphere.distinct("properties.lib_zone")
```

```
[ ]: %%js  
[  
    "Aubière",  
    "Aulnat",  
    "Beaumont",  
    "Blanzat",  
    "Ceyrat",  
    "Chamalières",  
    "Châteaugay",  
    "Clermont-Ferrand",  
    "Cournon-d'Auvergne",  
    "Cébazat",  
    "Durtol",  
    "Gerzat",  
    "Le Cendre",  
    "Lempdes",  
    "Nohanent",  
    "Orcines",  
    "Pont-du-Château",  
    "Pérignat-lès-Sarliève",  
    "Romagnat",  
    "Royat",  
    "Saint-Genès-Champanelle"  
]
```

4- Quelle est le code qualité moyen par commune ?

```
[ ]: db.Atmosphere.aggregate([  
    {  
        $group: {  
            _id: "$properties.lib_zone",  
            qualite_moyen: { $avg: "$properties.code_qual" }  
        }  
    }  
])
```

```
[4]: [{'_id': 'Beaumont', 'qualite_moyen': 2},  
      {'_id': 'Cébazat', 'qualite_moyen': 2},  
      {'_id': 'Le Cendre', 'qualite_moyen': 2},  
      {'_id': 'Aubière', 'qualite_moyen': 2},
```

```
{'_id': 'Durtol', 'qualite_moyen': 2.111111111111111},
{'_id': 'Orcines', 'qualite_moyen': 2.111111111111111},
{'_id': 'Pérignat-lès-Sarliève', 'qualite_moyen': 2},
{'_id': 'Aulnat', 'qualite_moyen': 2},
{'_id': 'Ceyrat', 'qualite_moyen': 2.111111111111111},
{'_id': 'Châteaugay', 'qualite_moyen': 2},
{'_id': 'Gerzat', 'qualite_moyen': 2},
{'_id': 'Nohanent', 'qualite_moyen': 2},
{'_id': 'Chamalières', 'qualite_moyen': 2.111111111111111},
{'_id': 'Cournon-d'Auvergne', 'qualite_moyen': 2},
{'_id': 'Lempdes', 'qualite_moyen': 2},
{'_id': 'Royat', 'qualite_moyen': 2.111111111111111},
{'_id': 'Clermont-Ferrand', 'qualite_moyen': 2},
{'_id': 'Blanzat', 'qualite_moyen': 2},
{'_id': 'Saint-Genès-Champanelle', 'qualite_moyen': 2.111111111111111},
{'_id': 'Pont-du-Château', 'qualite_moyen': 2}]
```

5- Quelle est la qualité de l'air la plus récente dans un rayon de 1 km de la position GPS 45.77441539864761, 3.0890499134717686

```
[ ]: db.Atmosphere.createIndex({ "geometry.coordinates": "2dsphere" })
db.Atmosphere.aggregate([
  {
    $geoNear: {
      near: {
        type: "Point",
        coordinates: [3.0890499134717686, 45.77441539864761]
      },
      distanceField: "distance",
      maxDistance: 1000,
      spherical: true
    }
  },
  {
    $sort: {
      "properties.date_ech": -1
    }
  },
  {
    $limit: 1
  },
  {
    $project: {
      _id: 0,
      "properties.code_qual": 1,
      "properties.date_ech": 1,
      "properties.code_zone": 1,
    }
  }
])
```

```

        "properties.lib_zone":1
    }
}
])

```

```

[8]: %%js
[
  {
    "properties": {
      "date_ech": "2023-10-25T00:00:00Z",
      "code_qual": 2,
      "code_zone": "63113",
      "lib_zone": "Clermont-Ferrand"
    }
  }
]

```

```

[8]: [{ 'properties': { 'date_ech': '2023-10-25T00:00:00Z',
  'code_qual': 2,
  'code_zone': '63113',
  'lib_zone': 'Clermont-Ferrand' }}]

```

Combien y-a t'il de relevé d'indice différent pour les communes dont la lettre commence par C

```

[ ]: db.Atmosphere.aggregate([
  {
    $match: {
      "properties.lib_zone": { $regex: /^C/i }
    }
  },
  {
    $group: {
      _id: "$properties.lib_zone",
      count: { $sum: 1 }
    }
  },
  {
    $group: {
      _id: null,
      Nombre_total: { $sum: 1 }
    }
  }
])

```

```

[5]: %%js
[
  {
    "_id": null,

```

```

    "Nombre_total": 6
  }
]

```

<IPython.core.display.Javascript object>

7. Pour chaque libellé de qualité de l'air, quelles sont les communes qui lui sont associé ?

```

[ ]: db.Atmosphere.aggregate([
    {
      $group: {
        _id: "$properties.lib_qual",
        communes: { $addToSet: "$properties.lib_zone" }
      }
    }
  ])

```

```

[ ]: [
    {
      "_id": "Moyen",
      "communes": [
        "Saint-Genès-Champanelle",
        "Durtol",
        "Blanzat",
        "Pont-du-Château",
        "Le Cendre",
        "Cournon-d'Auvergne",
        "Orcines",
        "Pérignat-lès-Sarliève",
        "Cébazat",
        "Châteaugay",
        "Ceyrat",
        "Beaumont",
        "Chamalières",
        "Lempdes",
        "Aulnat",
        "Gerzat",
        "Nohanent",
        "Aubière",
        "Clermont-Ferrand",
        "Romagnat",
        "Royat"
      ]
    },
    {
      "_id": "Dégradé",
      "communes": [
        "Royat",

```

```

        "Ceyrat",
        "Durtol",
        "Chamalières",
        "Orcines",
        "Saint-Genès-Champanelle"
    ]
}
]
```

8. Quelle est la concentration moyenne de PM10 dans la région Auvergne-Rhône-Alpes le 22 octobre 2023 ?

```

[ ]: db.Atmosphere.aggregate([
  {
    $match: {
      "properties.epsg_reg": "2154",
      "properties.date_ech": { $gte: "2023-10-22T00:00:00Z", $lt:
↪ "2023-10-23T00:00:00Z" },
      "properties.code_pm10": { $exists: true }
    }
  },
  {
    $group: {
      _id: null,
      avg_pm10_auv_rhone_alpes: { $avg: "$properties.conc_pm10" }
    }
  }
])
```

```

[ ]: %%js
[
  {
    "_id": null,
    "avg_pm10_auv_rhone_alpes": 7.739757142857142
  }
]
```

9. Afficher l'évolution de la concentration de NO2 dans la commune de Clermont-Ferrand à travers le temps.

```

[ ]: db.Atmosphere.find({
  "properties.lib_zone": "Clermont-Ferrand",
  "properties.code_no2": { $exists: true }
}, {
  "_id": 0,
  "properties.date_ech": 1,
  "properties.conc_no2": 1
}).sort({ "properties.date_ech": 1 })
```

```

[ ]: %%js
[
  {
    "properties": {
      "date_ech": "2023-10-17T00:00:00Z",
      "conc_no2": 86.3478
    }
  },
  {
    "properties": {
      "date_ech": "2023-10-17T00:00:00Z",
      "conc_no2": 86.3478
    }
  },
  {
    "properties": {
      "date_ech": "2023-10-18T00:00:00Z",
      "conc_no2": 44.4089
    }
  },
  {
    "properties": {
      "date_ech": "2023-10-18T00:00:00Z",
      "conc_no2": 44.4089
    }
  },
  {
    "properties": {
      "date_ech": "2023-10-19T00:00:00Z",
      "conc_no2": 56.8136
    }
  },
  {
    "properties": {
      "date_ech": "2023-10-19T00:00:00Z",
      "conc_no2": 56.8136
    }
  },
  {
    "properties": {
      "date_ech": "2023-10-20T00:00:00Z",
      "conc_no2": 48.7794
    }
  },
  {
    "properties": {
      "date_ech": "2023-10-20T00:00:00Z",

```



```

        "conc_no2": 48.7794
    },
    {
        "properties": {
            "date_ech": "2023-10-21T00:00:00Z",
            "conc_no2": 10.6188
        }
    },
    {
        "properties": {
            "date_ech": "2023-10-21T00:00:00Z",
            "conc_no2": 10.6188
        }
    },
    {
        "properties": {
            "date_ech": "2023-10-22T00:00:00Z",
            "conc_no2": 55.7
        }
    },
    {
        "properties": {
            "date_ech": "2023-10-22T00:00:00Z",
            "conc_no2": 55.7
        }
    },
    {
        "properties": {
            "date_ech": "2023-10-23T00:00:00Z",
            "conc_no2": 48.505
        }
    },
    {
        "properties": {
            "date_ech": "2023-10-23T00:00:00Z",
            "conc_no2": 48.505
        }
    },
    {
        "properties": {
            "date_ech": "2023-10-24T00:00:00Z",
            "conc_no2": 51.4811
        }
    },
    {
        "properties": {

```

```

        "date_ech": "2023-10-24T00:00:00Z",
        "conc_no2": 51.4811
    }
},
{
    "properties": {
        "date_ech": "2023-10-25T00:00:00Z",
        "conc_no2": 49.5019
    }
},
{
    "properties": {
        "date_ech": "2023-10-25T00:00:00Z",
        "conc_no2": 49.5019
    }
}
]

```

```

[7]: import pandas as pd
data = [
    {"properties": {"date_ech": "2023-10-17T00:00:00Z", "conc_no2": 86.3478}},
    {"properties": {"date_ech": "2023-10-17T00:00:00Z", "conc_no2": 86.3478}},
    {"properties": {"date_ech": "2023-10-18T00:00:00Z", "conc_no2": 44.4089}},
    {"properties": {"date_ech": "2023-10-18T00:00:00Z", "conc_no2": 44.4089}},
    {"properties": {"date_ech": "2023-10-19T00:00:00Z", "conc_no2": 56.8136}},
    {"properties": {"date_ech": "2023-10-19T00:00:00Z", "conc_no2": 56.8136}},
    {"properties": {"date_ech": "2023-10-20T00:00:00Z", "conc_no2": 48.7794}},
    {"properties": {"date_ech": "2023-10-20T00:00:00Z", "conc_no2": 48.7794}},
    {"properties": {"date_ech": "2023-10-21T00:00:00Z", "conc_no2": 10.6188}},
    {"properties": {"date_ech": "2023-10-21T00:00:00Z", "conc_no2": 10.6188}},
    {"properties": {"date_ech": "2023-10-22T00:00:00Z", "conc_no2": 55.7}},
    {"properties": {"date_ech": "2023-10-22T00:00:00Z", "conc_no2": 55.7}},
    {"properties": {"date_ech": "2023-10-23T00:00:00Z", "conc_no2": 48.505}},
    {"properties": {"date_ech": "2023-10-23T00:00:00Z", "conc_no2": 48.505}},
    {"properties": {"date_ech": "2023-10-24T00:00:00Z", "conc_no2": 51.4811}},
    {"properties": {"date_ech": "2023-10-24T00:00:00Z", "conc_no2": 51.4811}},
    {"properties": {"date_ech": "2023-10-25T00:00:00Z", "conc_no2": 49.5019}},
    {"properties": {"date_ech": "2023-10-25T00:00:00Z", "conc_no2": 49.5019}},
]

```

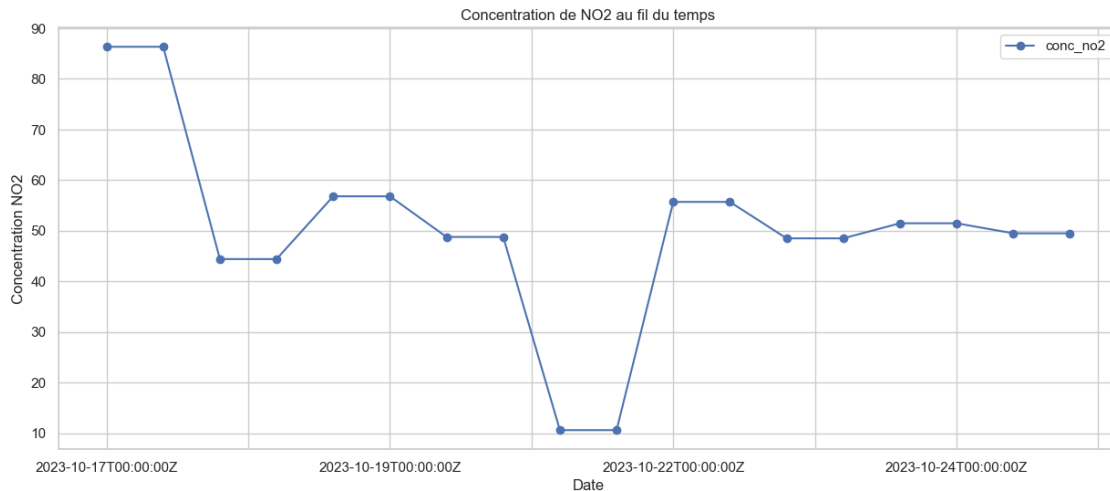
```
df = pd.DataFrame([entry['properties'] for entry in data])
```

```

[15]: import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(15, 6))
df.plot(x='date_ech', y='conc_no2', ax=ax, marker='o', linestyle='-')
plt.title('Concentration de NO2 au fil du temps')
plt.xlabel('Date')

```

```
plt.ylabel('Concentration NO2')
plt.show()
```



10. Quelle est la commune qui a l'indice de concentration d'ozone le plus faible sur la période observée ?

```
[ ]: db.Atmosphere.find({
    "properties.conc_o3": { $exists: true }
}, {
    "_id": 0,
    "properties.lib_zone": 1, "properties.conc_o3": 1
}).sort({ "properties.conc_o3": 1 }).limit(1)
```

```
[ ]: [
  {
    "properties": {
      "lib_zone": "Châteaugay",
      "conc_o3": 54.2012
    }
  }
]
```

11. Quelle serait la clé de sharding à utiliser pour pouvoir s'assurer que tous les enregistrements d'une même commune soient traités par le même nœud ?

La clé de sharding idéale pour s'assurer que tous les enregistrements d'une même commune soient traités par le même nœud serait le champ "properties.lib_zone" (ou le code postal), qui représente le nom de la commune. En utilisant ce champ comme clé de sharding, MongoDB s'assurera que tous les documents ayant la même valeur dans le champ "properties.lib_zone" seront stockés sur le même nœud de shard.