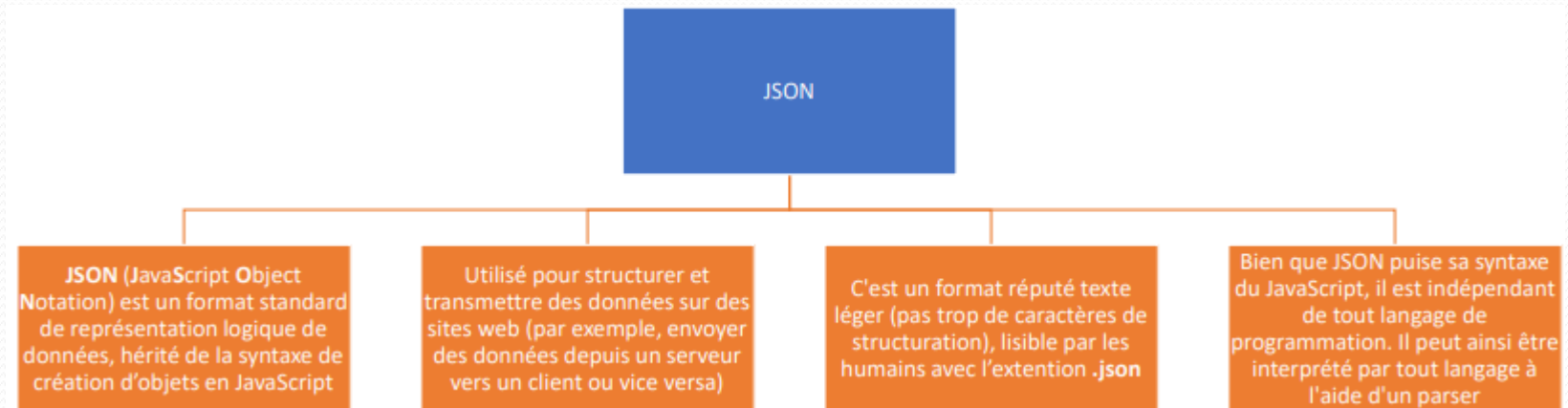


Modéliser les documents

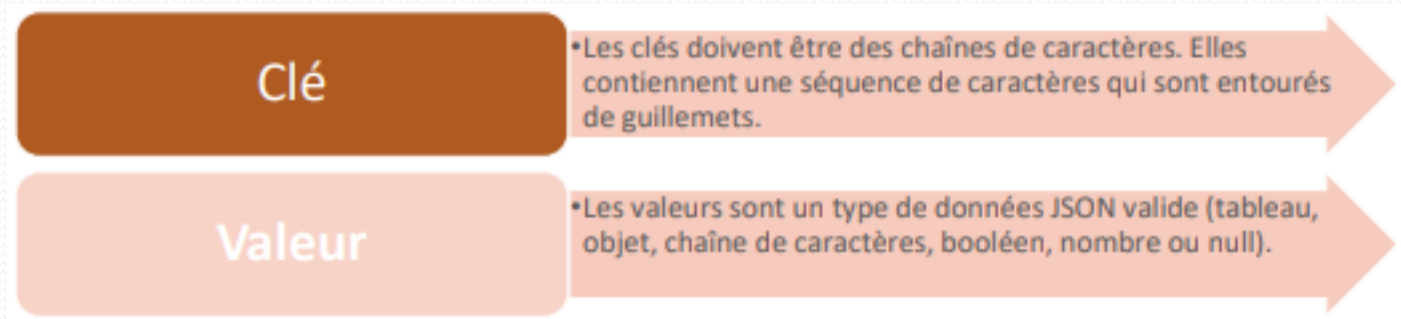
(MongoDB)

Structurer un document JSON



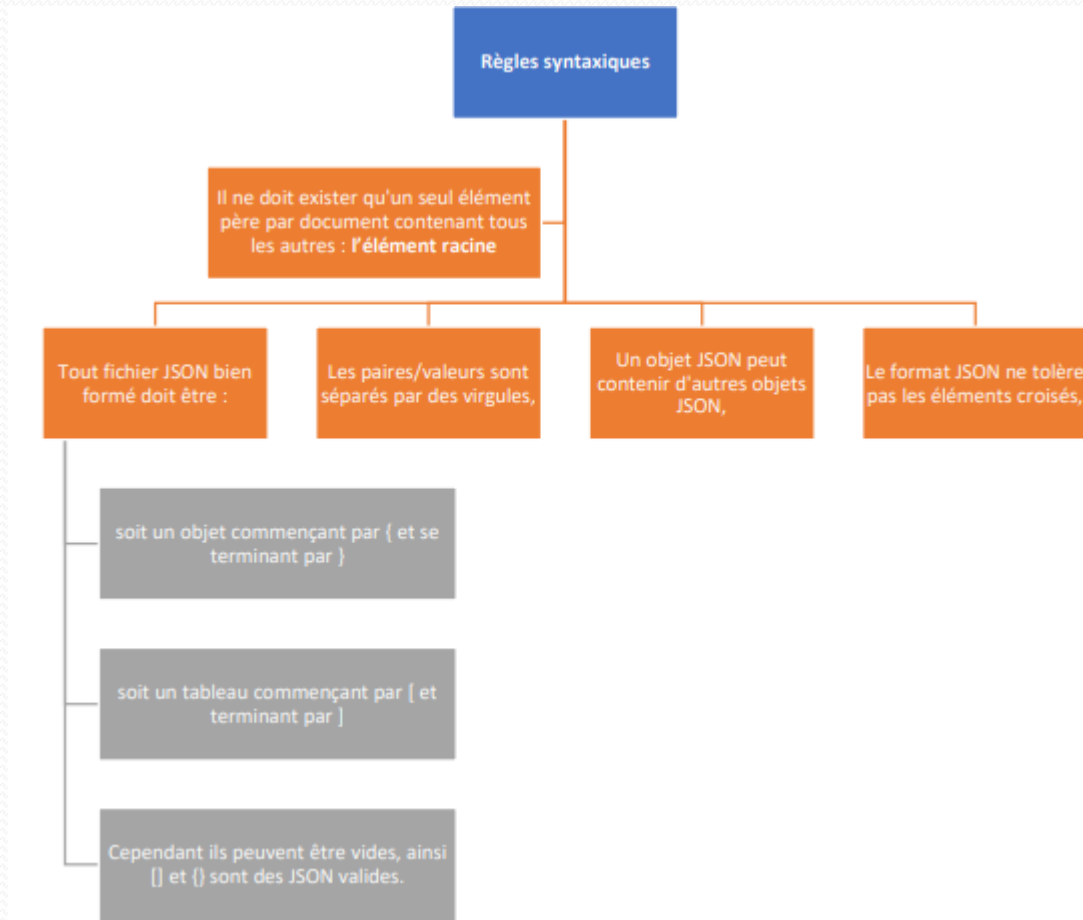
Objet JSON

- Un objet JSON se base sur deux éléments essentiels : Les clés et les valeurs.



- Un objet JSON commence et se termine par des accolades {}.
- Il peut contenir plusieurs paires clé/valeur, séparées par une virgule., La clé est suivie de « : » pour la distinguer de la valeur.

Règles syntaxiques



Exemple de fichier JSON

```
{
  [
    { _id: 12,
      "prenom": "Kamal",
      "type": "Stagiaire",
      "filiere": "Dev Digital",
      "groupe": "DD203",
      "niveau": "2A",
      "option": "Mobile"},
    {
      _id: 17,
      "prenom": "Amina",
      "filiere": "Infra Digitale ",
      "niveau": "1A",
    }
  ]
}
```

Les Types de données

- Dans MongoDB, les documents sont stockés dans **BSON**, le format codé binaire de JSON, elle prend en charge divers types de données à savoir:

- **String** : chaîne de caractères, les strings BSON sont en UTF-8,

```
{"_id":"1234", "prenom":"Amina"}
```

- **Entier**: entier qui peut etre stocker le type de données entier sous deux formes : entier signé 32 bits et entier signé 64 bits.

```
{"_id":"1234", "prenom":"Amina", "age": 19}
```

- **Double** : utilisé pour stocker les valeurs à virgule flottante.

```
{"_id":"1234", "prenom":"Amina", "moyBaccalaureat":14.25}
```

Les Types de données

- **Boolean** : utilisé pour stocker vrai ou faux

```
{"_id":"1234","prenom":"Amina","moyBaccalaureat":14.25,"admis":true}
```

- **Date** :

- stocke la date sous forme de millisecondes (entier 64bits),
- La date peut être exprimée sous forme de string : `Date()` ou d'objet date `new Date()`

```
{  
  "prenom": "Ahmed",  
  "niveau": "1A",  
  "filiere": "Dev digital",  
  "DateInscription_1": Date(),  
  "DateInscription_2": new Date()  
}
```

Vue JSON

```
{  
  "_id" : ObjectId("62dd1dffa19f3d7ecc66252"),  
  "prenom" : "Ahmed",  
  "niveau" : "1A",  
  "filiere" : "Dev digital",  
  "DateInscription_1" : "Sun Jul 24 2022 11:25:03 GMT+0100",  
  "DateInscription_2" : ISODate("2022-07-24T10:25:03.613+0000")  
}
```

Les Types de données

- **Null** : utilisé pour stocker la valeur null,

```
{
  "_id": "1234",
  "prenom": "Amina",
  "telephone": null
}
```

- **Données binaires** : utilisé pour stocker les données binaires,

```
{
  "_id": "1234",
  "prenom": "Amina",
  "telephone": null,
  "BinaryValues": "10010001",
}
```


Les Types de données

- **Array:** Ensemble des valeurs pouvant être de types de données identiques ou différentes. Dans MongoDB, le array est créé à l'aide de crochets ([]).

```
{
  "_id": "1234",
  "prenom": "Kamal",
  "niveau": "2A",
  "option": "Mobile",
  "skills": [
    "python",
    "javascript",
    "php" ]
}
```

- **ObjectId (Id d'objet):** pour chaque nouveau document créé dans une collection, MongoDB crée automatiquement un identifiant d'objet unique `_id` s'il n'est pas créé explicitement.

```
{"prenom": "Ahmed", "niveau": "1A", "filier": "Dev digital"}
{"_id": 1234, "prenom": "Alaa", "niveau": "2A", "filier": "Dev digital"}
```

Vue JSON

```
{
  "_id" :
  ObjectId("62dd128a1a19f3d7ecc6624f"),
  "prenom" : "Ahmed",
  "niveau" : "1A",
  "filier" : "Dev digital"
}
{
  "_id" : 1234.0,
  "prenom" : "Alaa",
  "niveau" : "2A",
  "filier" : "Dev digital"
}
```

Comparaison entre base relationnelle et orientée document

Base de données relationnelle	MongoDB (Base de données orientée documents)
Base de données	Base de données
Table	Collection
Enregistrement	Document
Schéma de données fixe	Schéma de données flexible
Les enregistrements de la table doivent avoir le même ensemble de champs	Les documents d'une même collection n'ont pas besoin d'avoir le même ensemble de champs
Le type de données d'un champ est fixe pour tous les enregistrements de la table	Le type de données d'un champ peut différer d'un document à l'autre d'une collection.

- Un document est défini sous la forme d'un objet JSON

Les types de relations entre les données sous MongoDB

- MongoDB détermine deux types de relations entre les données :
 1. **Les relations d'imbrication (embedding) :**
 - L'imbrication d'un (ou d'une partie) d'un document dans un autre, on parle de document autonome,
 - L'imbrication permet d'éviter de faire des jointures: inutile de faire des jointures pour restituer l'information qui n'est pas dispersée sur plusieurs entités (tables en relationnel),
 - On utilise l'imbrication des documents (embedding) quand les documents sont très petits et n'ont pas tendance à grandir dans le futur. La taille des documents ne doit pas dépasser 16Mb,
 - Adéquates pour les contextes qui privilégient la recherche à la mise à jour,

Les relations d'imbrication

- Pour une relation entre deux documents A et B, cela consiste à imbriquer partiellement ou totalement le document B dans le document A.
- Ce modèle de relations dénormalisés permet aux applications de récupérer et de manipuler des données associées en une seule opération de base de données,
- Exemple :

```
{
  "_id": "1234",
  "prenom": "Amina",
  "filier": {
    "_id": "DD",
    "intitule": "Développement digital"
  }
}
```

} Document imbriqué

Les types de relations entre les données sous MongoDB

2. Les relations de liaisons (Linking) :

- La duplication de l'identifiant d'un document dans un autre document,
- Reprend, en quelque sorte, le concept de jointure entre les tables relationnelles,
- N'est privilégiée dans le contexte de relations plusieurs-plusieurs:
 - Exemple : Commande → Produit
 - Un produit peut être commandé plusieurs fois et une commande peut contenir plusieurs produits, Une imbrication des Produits dans la commande aura de gros impacts sur les mises à jour (tous les produits à mettre à jour !)

Les relations de liaison (linking)

- Les relations de liaison permettent d'inclure les liens ou des références d'un document dans un autre,
- Ce modèle de relations récupère les données en deux étapes:
 - une première requête pour récupérer l'identifiant,
 - une deuxième requête pour récupérer les données de l'autre côté de la relation.
- **Exemple :**

```
{ "_id": "DD", "intitule": "développement digital" }  
{  
  "_id": "1234",  
  "prenom": "Amina",  
  "filiere": "DD"  
}
```