

Base de données NoSQL

Un nouveau paradigme

M. Sacha RESTOUEIX
Formateur DWM
AFPA de Brive la Gaillarde

Objectifs Pédagogiques

- Se familiariser au nouveau paradigme des bases de données **NoSQL**
- Maîtriser les concepts autour du **NoSQL**
- Construire/manipuler une base **NoSQL** avec **MongoDB (v 6.0)**

Introduction

Depuis le début des années 2000, il y a une forte augmentation des données sur le Web, notamment avec l'émergence des réseaux sociaux et de la popularité croissante « *d'Internet* » (Facebook, X (anciennement Twitter), Google, YouTube, Amazon etc...)

De tels volumes de données ne sont plus gérables avec des solutions dites traditionnelles

Cela a conduit à une nouvelle façon de gérer les données :

Émergence des bases de données NoSQL (**N**ot **O**nly **S**QL) qui sont non relationnelles.

NoSQL

Les bases de données **NoSQL** apparaissent autours des années 2010.

C'est une famille de base de données qui s'écarte du schéma classique des bases de données relationnelles (**SGBDR**)

Objectifs

- Se soustraire aux contraintes lourdes des **SGBDR**
- **Dénormaliser** les modèles (dupliquer la donnée là où elle est demandée)
- Favoriser les **performances**
- **Moins couteux** (ajout de serveurs)

NoSQL

Les Principes du NoSQL

Dans un **SGBDR**, les *transactions* sont dites **ACID**

- ➔ **Atomicité** : Toutes les opérations d'une *transaction* sont effectuées, autrement aucune n'est réalisée.
- ➔ **Cohérence** : Le contenu de la base doit être cohérent du début à la fin.
- ➔ **Isolation** : Les modifications d'une *transaction* ne sont visibles que quand celle-ci a été validée.
- ➔ **Durabilité** : Une fois la *transaction* validée, l'état de la base est permanent (quel que soit l'incident qui survient, panne, coupure, etc.)

NoSQL

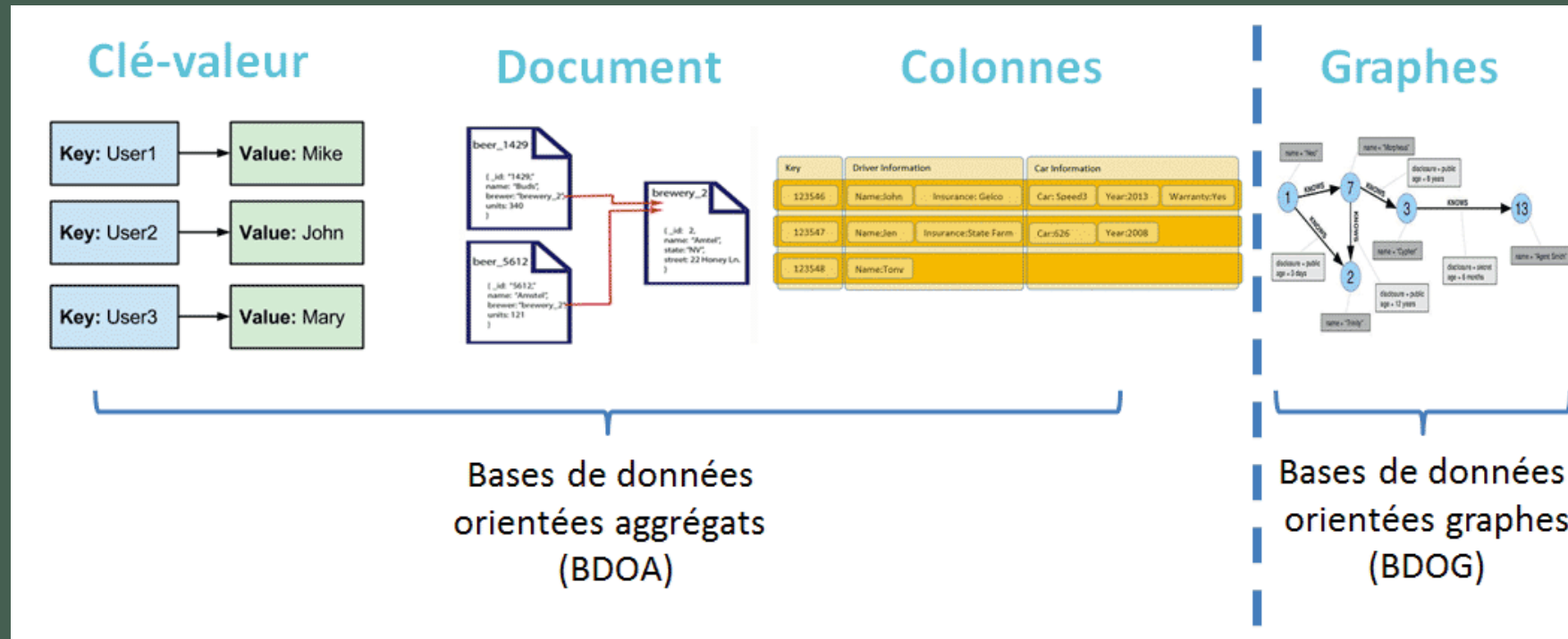
Les Principes du NoSQL

Les bases **NoSQL** sont moins contraignantes afin de garantir un haut niveau de performance. Elles sont dites **BASE**

- ➡ **Basically Available** : Le système garantit un taux de disponibilité de la donnée.
- ➡ **Soft-state** : La base n'est pas nécessairement cohérente à tout instant.
- ➡ **Eventually consistent** : La base atteindra, à terme, un état cohérent.

NOSQL : 4 types de bases de données

- ▶ Chaque type répond à un besoin spécifique.



Quelques noms (gratuits ou payants)

- ▶ Orientés clé-valeurs : Voldemort, Redis, DynamoDB, BerkeleyDB...
- ▶ Documents : MongoDB, CouchDB...
- ▶ Colonnes : Apache Cassandra, HBase, Google BigTable...
- ▶ Graphes : Neo4j, Titan...

MongoDB le plus célèbre des SGBD NoSQL

Dans le cadre de cette formation, nous allons étudier **MongoDB v 6.0**

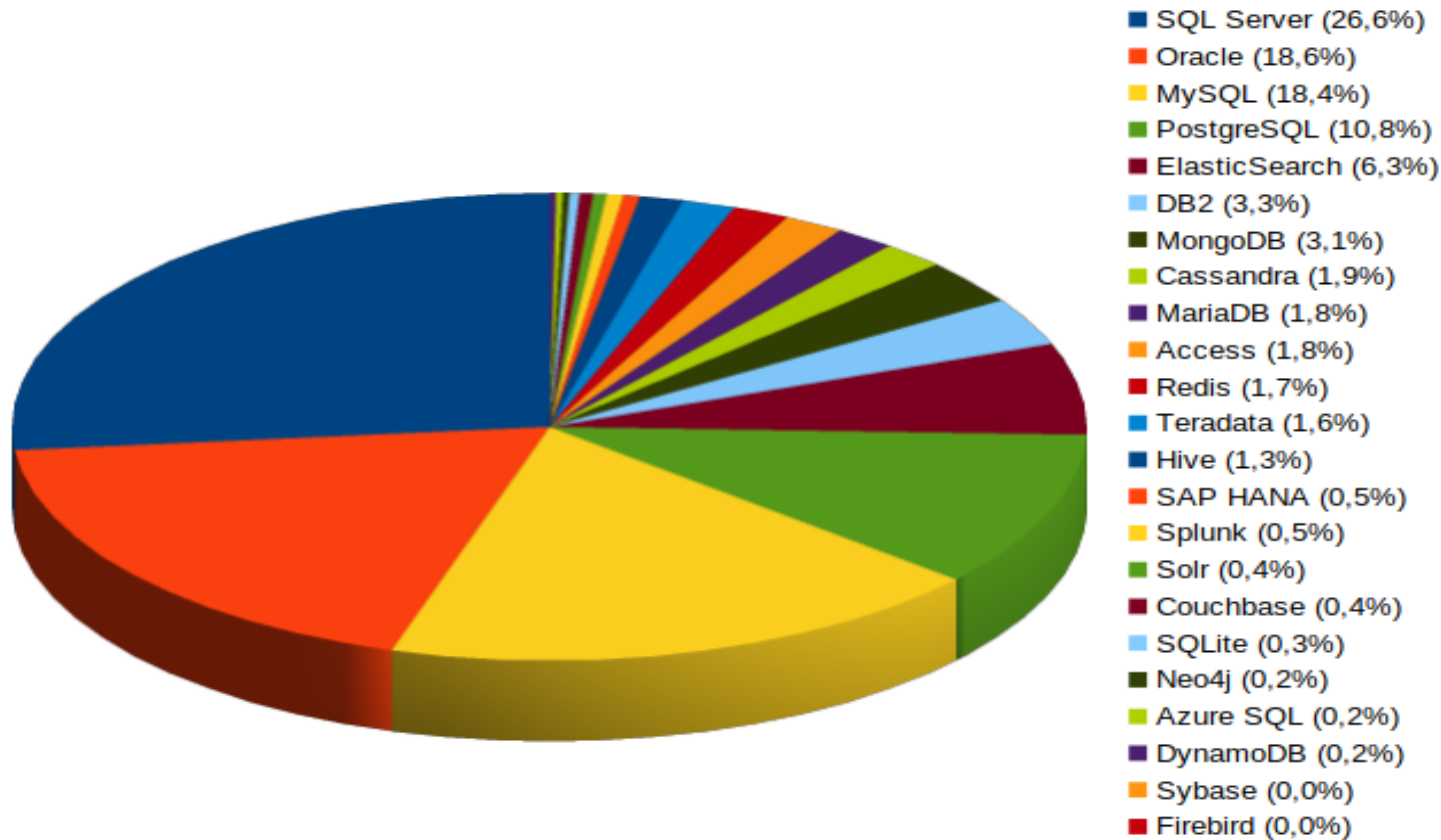
MongoDB est une base de données **NoSQL** orientée **document** qui stocke les données sous forme de **documents** au format **JSON** (paires clé/valeur). (En fait au format **BSON** qui est une *forme binaire* du **JSON**)

Il s'agit d'une base de données **Open Source** et **Multiplateforme** qui offre des performances et une évolutivité élevée.

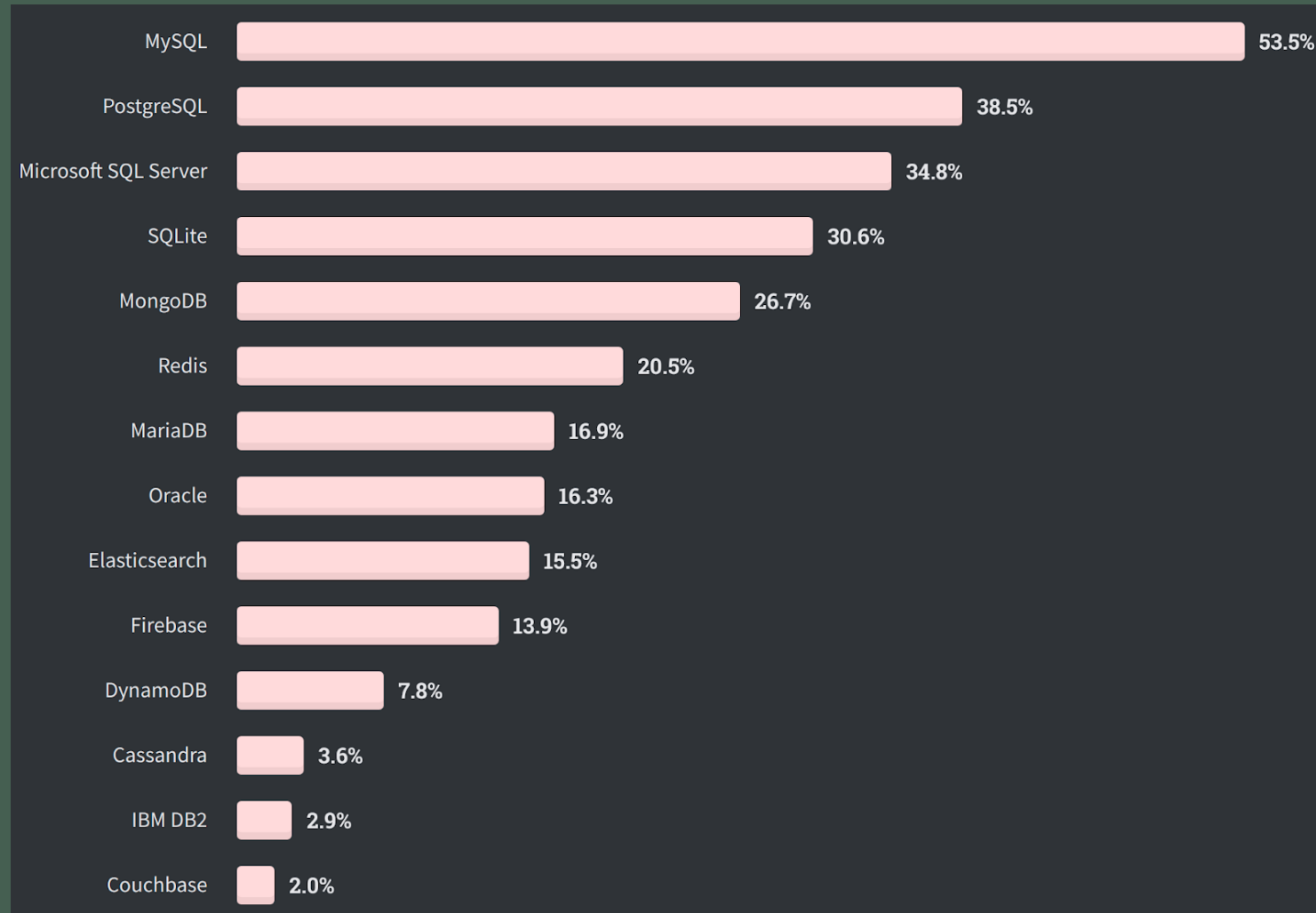
<https://www.mongodb.com/fr-fr>

Popularité des divers Bases de données au niveau des offres d'emploi

Popularité des SGBD dans les offres d'emploi en 2020



Popularité des divers Bases de données en production de projets - année 2020



MongoDB

- Base de données **NoSQL**
- Sortie en 2009
- Orientée document
- Pas de schéma, pas de structure de table, pas de typage
- Le Shell **Mongosh** fournit un langage basé sur **JavaScript**
- MongoDB prend en charge *la mise à l'échelle horizontale* via le « *sharding* » (ajout de serveurs)
<https://www.mongodb.com/docs/manual/sharding/>



Installation

3 étapes nécessaires

1 - Le Serveur

<https://www.mongodb.com/try/download/community>

serveur se lance via l'exécutable **mongod.exe**

2 - Un répertoire stockant les données
par défaut sous Windows : C:/data/db

3 - Une interface

Un interpréteur de commande **mongo.exe**

Ou

Une interface graphique : **MongoDB Compass (v 1.37.0)**

<https://www.mongodb.com/products/compass>

Dénormalisation

L'information est modélisée au « *format JSON* »
Nécessite de **dénormaliser** les tables traditionnelles



id	titre
1	Her
2	Avengers

film_id	acteur_id
1	1
1	2
2	1

id	nom	prenom
1	Johansson	Scarlett
2	Phoenix	Joaquin

```
1 { "_id": "Her", "acteurs": [{ "nom": "Johansson", "prenom": "Scarlett" }, { "nom": "Phoenix",  
  prenom": "Joaquin" }] }  
2 { "_id": "Avengers", "acteurs": [{ "nom": "Johansson", "prenom": "Scarlett" }] }
```

Base de données
relationnelle classique
Normalisée

Base de données NoSQL
Dénormalisée (pas de FK)

Parallèle entre SGBDR et MongoDB

Le tableau suivant montre la relation entre la terminologie de SGBDR et MongoDB.

SGBDR	MongoDB
Base de données	Base de données
Table	Collection
Tuple	Document
Colonne	Champ
Jointure	Document incorporé
Clé primaire	Clé primaire (Clé par défaut _id fournie par mongodb elle-même)

MongoDB



Créer/utiliser une Base de données

use bdFilm // Va créer la Base de données bdFilm si elle n'existe pas déjà. Cette Bdd sera constituée de une ou plusieurs **collections**.

Créer une collection

db.createCollection(« acteurs ») // Va créer la collection *acteurs*, elle sera constituée de **documents** apparentés à des objets JSON.

Supprimer une collection

db.acteurs.drop() // Va supprimer la collection acteurs

Interroger / Manipuler la base

Une action sur une **collection** doit être préfixée du nom de la collection
db.<nom Collection>.<action à réaliser>

Action

Insérer (**insert**) // Avec ses variantes **insertOne** et **insertMany**

Lire (**find**) // Avec ses variantes **findOne**, **findOneAndDelete**, ...

Mettre à jour (**update**) // Avec ses variantes **updateOne**, **updateMany**
et **replaceOne**

Supprimer (**delete**) // Avec ses variantes **deleteOne**, **deleteMany**, ...

Interroger / Manipuler la base

Insérer un **document**, commande générique :
`db.<nom Collection>.insert(<document>)`

Exemple

```
db.acteurs.insertOne(  
  { prenom : « Morena »,  
    nom : « BACCARIN » })
```

Dans une collection, les documents n'ont pas nécessairement la même structure

Exemple

```
db.acteurs.insertOne(  
  { prenom : « Chris »,  
    nom : « PRATT »,  
    naissance : « 21/06/1979 » })
```

MongoDB

Interroger / Manipuler la base

sélectionner un document
`db.<nom Collection>.find(<critère>)`

*Remarque : C'est l'équivalent de la clause **SELECT** en **SQL***

Exemples de ses variantes

Sélectionner un enregistrement
`db.acteurs.findOne()`

Sélectionner tous les enregistrements
`db.acteurs.find({})`



MongoDB



mongoDB



mongoDB Compass

Interroger / Manipuler la base

Restriction sur une sélection

Commande générique :

```
db.<nom Collection>.find(<critère>)
```

Exemples

Avec un critère

```
db.acteurs.find({prenom : « Chris »})
```

Avec plusieurs critères

```
db.acteurs.find({ prenom : « Chris »,  
                 nom : « PRATT » })
```

Avec objets imbriqués

```
db.acteurs.find({ prenom : « Chris »,  
                 nom : « PRATT »,  
                 ville.codePostal : «19240 »})
```

MongoDB



Interroger / Manipuler la base

Restriction sur une sélection (projection sur une colonne)

Conserver certaines colonnes dans l'affichage du résultat

```
db.<nom Collection>.find(<critère>, <colonne>)
```

Exemple

```
db.acteurs.find({  
    prenom : « Chris »,  
    nom : « PRATT »  
},  
{ nom : true }) // On peut aussi formuler nom : 1
```

MongoDB



Interroger / Manipuler la base

Restriction sur une sélection

Compter le nombre d'enregistrements

```
db.<nom Collection>.find(<critère>, <colonne>).count()
```

Exemple

```
db.acteurs.find({  
    prenom : « Chris »,  
    nom : « PRATT »  
},  
{ nom : true }).count()
```

MONGODB VS SQL

MySQL

```
INSERT INTO users (user_id, age, status)
VALUES ('bcd001', 45, 'A')
```

```
SELECT * FROM users
```

```
UPDATE users SET status = 'C'
WHERE age > 25
```

MongoDB

```
db.users.insert({
  user_id: 'bcd001',
  age: 45,
  status: 'A'
})
```

```
db.users.find()
```

```
db.users.update(
  { age: { $gt: 25 } },
  { $set: { status: 'C' } },
  { multi: true }
)
```

MongoDB



Opérateurs mathématiques



\$gt, \$gte	>, ≥	Plus grand que (<i>greater than</i>)	"a": {"\$gt": 10}
\$lt, \$lte	<, ≤	Plus petit que (<i>less than</i>)	"a": {"\$lt": 10}
\$ne	≠	Différent de (<i>not equal</i>)	"a": {"\$ne": 10}
\$in, \$nin	∈, ∉	Fait parti de (ou ne doit pas)	"a": {"\$in": [10, 12, 15, 18]}
\$or	∨	OU logique	"a": {"\$or": [{"\$gt": 10}, {"\$lt": 5}]}
\$and	∧	ET logique	"a": {"\$and": [{"\$lt": 10}, {"\$gt": 5}]}
\$not	¬	Négation	"a": {"\$not": {"\$lt": 10}}
\$exists	∃	La clé existe dans le document	"a": {"\$exists": 1}
\$size		test sur la taille d'une liste (uniquement par égalité)	"a": {"\$size": 5}

MongoDB



Restriction sur une sélection

Exemples

```
db.acteurs.find({  
    naissance : { $lt : 1950},    // Dont la date de  
    naissance est inférieure strictement à 1950  
    {   nom : true }  
})
```

```
db.acteurs.find({  
    $or : [  
        {« prenom » : « Alan »},  
        {« ville.codePostal » : « 19240 » }]  
    })
```

MongoDB



L'agrégation avec MongoDB

Définition : Les opérations *d'agrégation* traitent les enregistrements de données et renvoient les **résultats calculés**.

Les documents entrent dans un *pipeline* éventuellement à *plusieurs étapes (stages)* qui transforme les documents en un **résultat agrégé**.

MongoDB



L'agrégation avec MongoDB

Parallèle entre les *clauses SQL* et les **opérateurs MongoDB** relatif à l'**agrégation**.

SQL	MongoDB
WHERE	\$match
GROUP BY	\$group
HAVING	\$match
SELECT	\$project
ORDER BY	\$sort
LIMIT	\$limit
SUM	\$sum
COUNT	\$count

MongoDB



L'agrégation avec MongoDB

Exemple : Pour connaître par genre (masculin ou féminin) la *moyenne d'âge* des acteurs de la collection.

```
db.acteurs.aggregate( [  
  { $group : { « _id » : « $genre »,  
               « moyenne_age » : { $avg : « $age »}  
            }  
  }  
])
```


MongoDB






L'agrégation avec MongoDB Compass





Documents **Aggregations** Schema Explain Plan Indexes Validation


Pipeline  \$group \$sort

Collation { locale: 'simple' }

Untitled - modified  SAVE  CREATE NEW  EXPORT TO LANGUAGE

Stage 1 \$group  A sample of the aggregated results

Stage 2 \$sort  A sample of the aggregated results

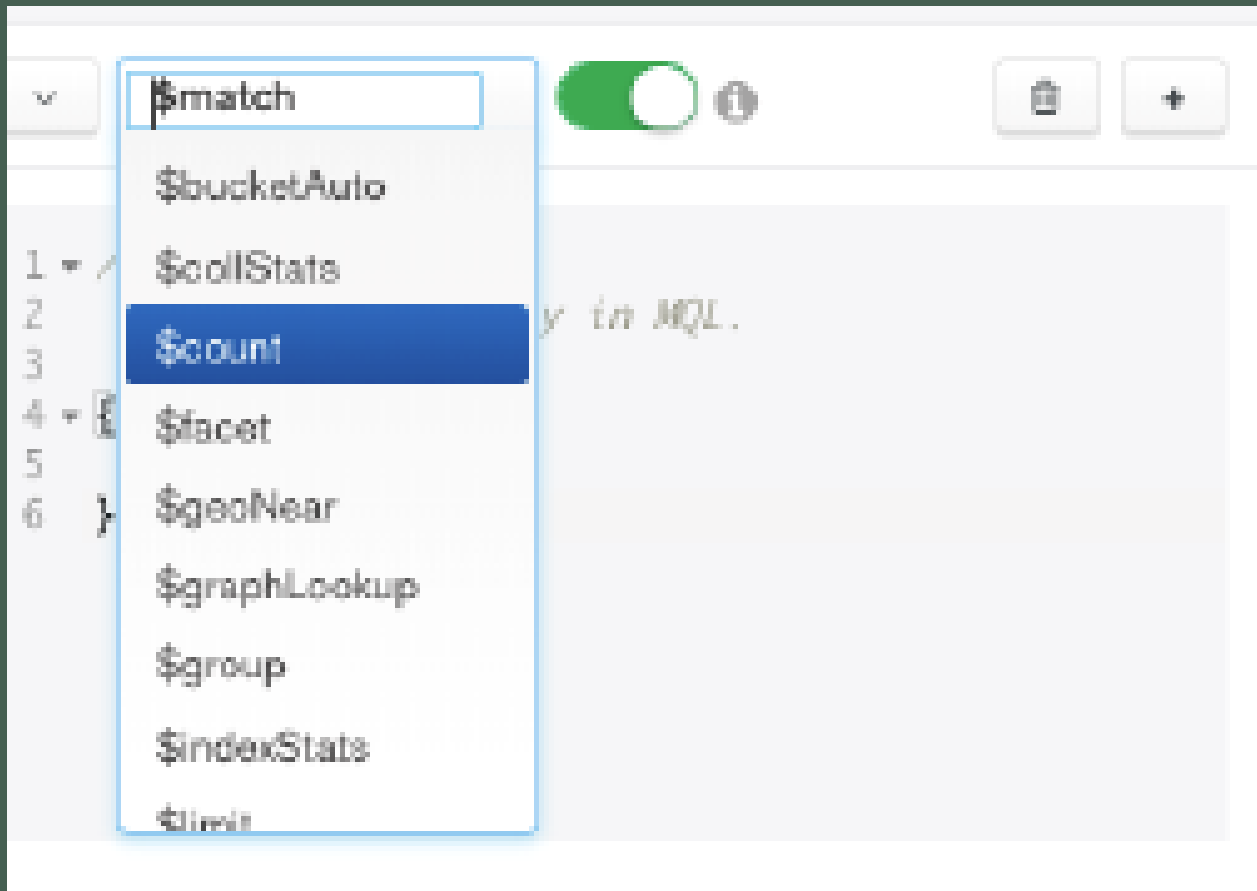
 Add Stage

Vous pouvez
construire un
pipeline
composé
éventuellement
de plusieurs
opérateurs
d'agrégation.

MongoDB



L'agrégation avec MongoDB Compass

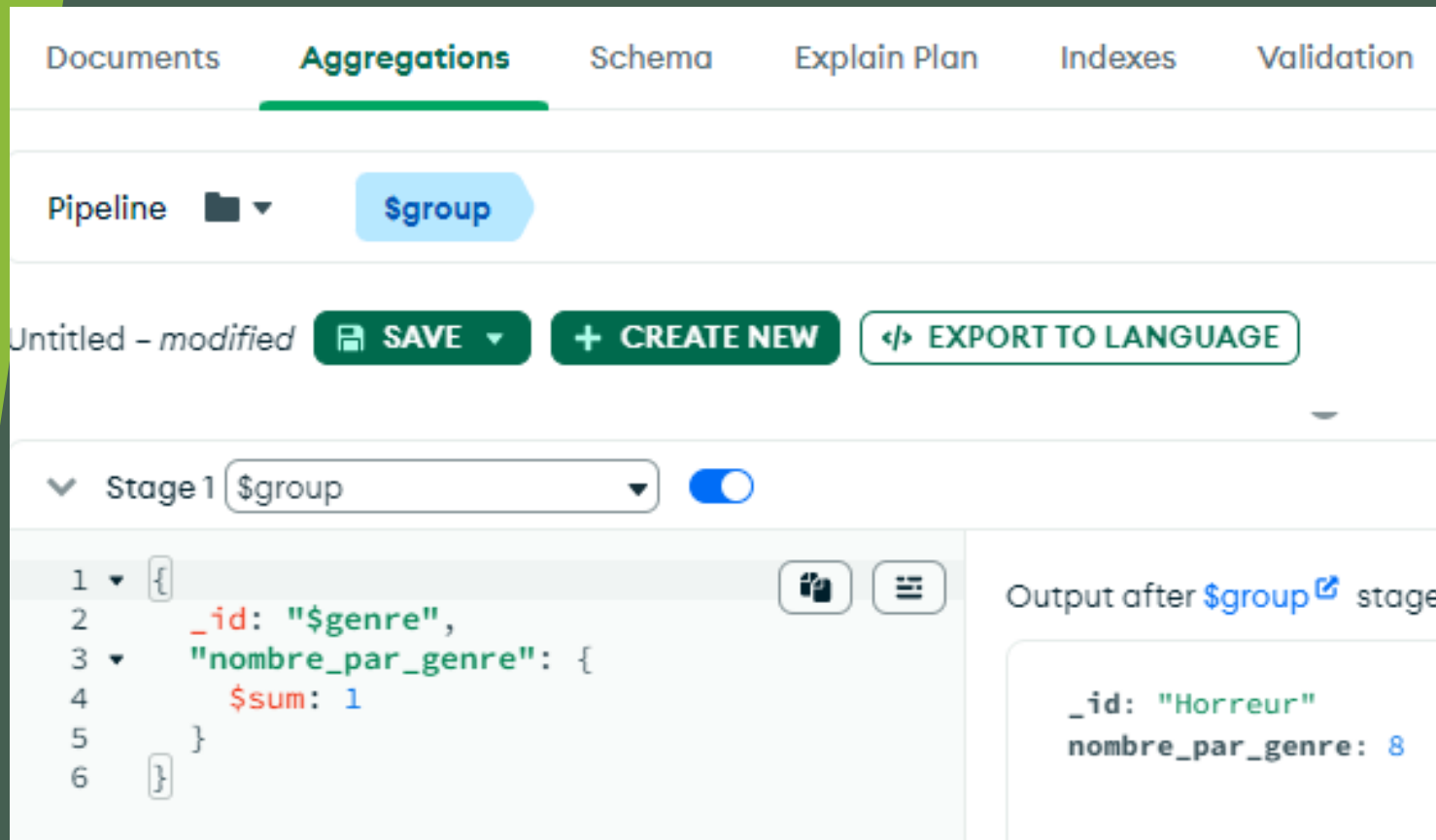


Pour chaque
opérateur
vous pouvez
définir le
champ et la
fonction
d'agrégation.

MongoDB



L'agrégation avec MongoDB Compass



L'interface proposée par *MongoDB Compass* est assez intuitive, elle permet d'obtenir de façon dynamique le résultat de la *requête d'agrégation* que vous aurez composé.

MongoDB



Mise à jour d'un document

Commande générique :

```
db.<nom Collection>.update(<critère>,<nouvelle valeur>)
```

Exemples de ses variantes

```
db.acteurs.updateOne({nom : « PRATT »} , { $set : {prenom :  
« Christopher »} } }
```

```
db.acteurs.updateMany({nom : « PRATT »} , { $set : {prenom :  
« Christopher »} } }
```


MongoDB



Mise à jour de la **structure** d'un document

Commande générique :

```
db.<nom Collection>.update(<critère>,<nouvelle valeur>)
```

Exemples

```
db.acteurs.updateMany(  
  { prenom : « Morena »,  
    nom : « BACCARIN »}, { $set : {note : 10}  
})
```

MongoDB



Suppression d'un document

Commande générique :

```
db.<nom Collection>.delete(<critère>,<nouvelle valeur>)
```

Exemples

```
db.acteurs.deleteOne({ nom : « BACCARIN »}) // Supprime le premier
```

```
db.acteurs.deleteMany({ nom : « BACCARIN »}) // Supprime tous
```

MongoDB



Pour en apprendre davantage sur la base de données MongoDB et sur tous l'écosystème qu'elle propose, je vous conseille les différentes lectures :

<https://www.mongodb.com/fr-fr>

<https://blog.dyma.fr/mongodb-la-base-de-donnees-nosql-la-plus-utilisee/>

<https://www.tutorialsteacher.com/mongodb>

CRÉDITS
OEUVRE COLLECTIVE DE L'AFPA

**Sous le pilotage de la DIIP
et du centre sectoriel Tertiaire**

EQUIPE DE CONCEPTION
M. RESTOUEIX Alexandre (Formateur)
Date de mise à jour : 08/07/2023
Date de dépôt légal : 2023

©AFPA 2023
Reproduction interdite