TP 1 MongoDB

Interagir avec les bases de données

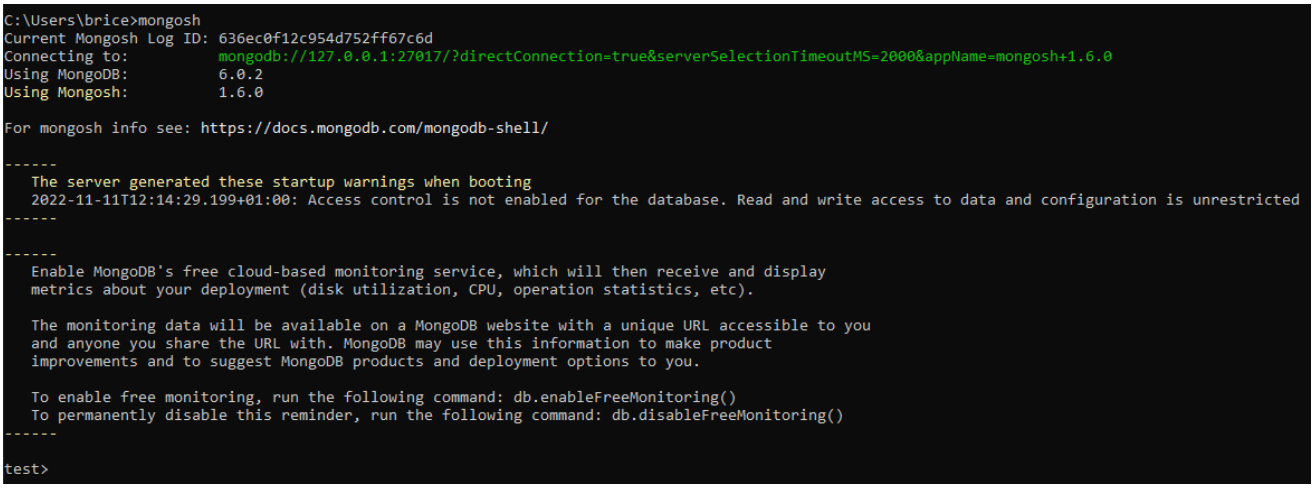
Avant de commencer ce TP, bien vouloir suivre les instructions ddocument **TP_1_Configurations**.

Local Database

Connexion à la base de données locale

```
$mongosh
```

La sortie attendue est la suivante:



Afficher la version du serveur

```
test> db.version()
6.0.2
```

Les basiques de MongoDB

Bases de données

Lister les bases de données

```
test> show dbs
admin    80.00 KiB
config  108.00 KiB
local    76.00 KiB
```

Connexion à une base de données existante

```
test> use admin
switched to db admin
admin>
```

> Lorsqu'on se connecte à une base de données, le texte affiché avant le curseur ">" change et porte désormais le nom de la base de données courante.

Connexion à une base de données inexistante

```
test> use myFirstDB
switched to db myFirstDB
myFirstDB>
```

> Cette commande va créer une nouvelle base de données, si celle-ci n'existe pas encore. Si elle existe, elle va commencer à l'utiliser.

TAF: Créer une base de données relatives au management d'une école

Collections

Créer des collections

Pour éffectuer des opérations dans la base de données courante, on utilise db.

```
esigManagement> db.createCollection("students")
{ ok: 1 }
```

Lister les collections

```
esigManagement> show collections
students
```

TAF: Créer une collection des enseignants de votre école et vérifier qu'elle a été créée.

Documents

```
esigManagement> db.students.insertOne({
 "firstName": "Encorvou",
 "lastName": "Ducobu",
 "email": "encorvou.ducobu@esigelec.com",
 "studentId": 20225454815
 })
{
  acknowledged: true,
  insertedId: ObjectId("636ec2a25418039f85c97412")
}
```

Nous venons d'ajouter l'élève Ducobu à notre base de données students.
Exécutez la commande suivante sans utiliser *db.createCollection*.

```
esigManagement> db.rooms.insertOne({"roomId": "B1215", "step": 3, "building": "B"})
```

TAF: Commenter le résultat de cette commande.
TAP: Une fois ce résultat commenté, supprimez la collection room en utilisant la méthode drop des collections.
TAF : Avec quels éléments d'une base relationnelle pourrait-on comparer une collection, un document

?

Nous pouvons en insérer plus et plusieurs d'un coup.

```
esigManagement> db.students.insertMany([
 {
 "firstName": "Son",
 "lastName": "Goku",
 "email": "son.goku@esigelec.com",
 "studentId": 20225454816
 },
 {
 "firstName": "Dora",
 "lastName": "exploratrice",
 "email": "dora.exploratrice@esigelec.com",
 "studentId": 20225454817
 }
 ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("636ec2de5418039f85c97413"),
    '1': ObjectId("636ec2de5418039f85c97414")
  }
}
```

Nous venons d'ajouter les élèves Goku et Dora dans notre collection **students.**

TAF: Ajoutez des enseignants. Pour chaque enseignant, on doit être capable de savoir:

- L'ancienneté
- Les enseignements
- Le salaire
- Le département
- Temps-partiel/plein

TAF: Ajoutez des informations sur la localisation des étudiants, leur promo(année), et leur dominante.

Opérations de base sur les documents dans MongoDB

Compter des documents

La fonction count de la collection students permet de compter le nombre de documents.

```
esigManagement> db.students.countDocuments()
3
```

Lister les documents

Pour trouver des documents, exécutez la commande suivante

```
esigManagement> db.students.find()
[
  {
    _id: ObjectId("636ec2a25418039f85c97412"),
```

```
    firstName: 'Encorvou',
    lastName: 'Ducobu',
    email: 'encorvou.ducobu@esigelec.com',
    studentId: 20225454815
  },
  {
    _id: ObjectId("636ec2de5418039f85c97413"),
    firstName: 'Son',
    lastName: 'Goku',
    email: 'son.goku@esigelec.com',
    studentId: 20225454816
  },
  {
    _id: ObjectId("636ec2de5418039f85c97414"),
    firstName: 'Dora',
    lastName: 'exploratrice',
    email: 'dora.exploratrice@esigelec.com',
    studentId: 20225454817
  }
]
```

Elle va lister tous les documents contenus dans la collection **students**.
TAF: Que remarquez vous dans les documents affichés?
TAF: Affichez les enseignants enregistrés.

Trier les documents

Afficher les élèves par ordre d'enregistrement dans la base. Du plus récent au plus ancien.

```
esigManagement> db.students.find().sort({"_id":-1})
[
  {
    _id: ObjectId("636ec2de5418039f85c97414"),
    firstName: 'Dora',
    lastName: 'exploratrice',
    email: 'dora.exploratrice@esigelec.com',
    studentId: 20225454817
  },
  {
    _id: ObjectId("636ec2de5418039f85c97413"),
    firstName: 'Son',
    lastName: 'Goku',
    email: 'son.goku@esigelec.com',
    studentId: 20225454816
  },
  {
    _id: ObjectId("636ec2a25418039f85c97412"),
    firstName: 'Encorvou',
    lastName: 'Ducobu',
    email: 'encorvou.ducobu@esigelec.com',
    studentId: 20225454815
  }
]
```

Comment sont identifiés uniquement les documents?

TAF: Afficher la liste des étudiants par ordre alphabétique

TAF: Afficher la liste des étudiants par ordre d'ancienneté(du plus ancien au dernier arrivé)

```
esigManagement> db.students.find().sort({"fisrstName":1})
[
  {
    _id: ObjectId("636ec2de5418039f85c97414"),
    fisrstName: 'Dora',
    lastName: 'exploratrice',
    email: 'dora.exploratrice@esigelec.com',
    studentId: 20225454817
  },
  {
    _id: ObjectId("636ec2a25418039f85c97412"),
    fisrstName: 'Encorvou',
    lastName: 'Ducobu',
    email: 'encorvou.ducobu@esigelec.com',
    studentId: 20225454815
  },
  {
    _id: ObjectId("636ec2de5418039f85c97413"),
    fisrstName: 'Son',
    lastName: 'Goku',
    email: 'son.goku@esigelec.com',
    studentId: 20225454816
  }
]
```

Limiter les documents

Afficher le premier étudiant inscrit à l'école

```
esigManagement> db.students.find().limit(1)
[
  {
    _id: ObjectId("636ec2a25418039f85c97412"),
    fisrstName: 'Encorvou',
    lastName: 'Ducobu',
    email: 'encorvou.ducobu@esigelec.com',
    studentId: 20225454815
  }
]
```

TAF: Afficher le dernier étudiant inscrit à l'école

Se déconnecter du serveur

Il est possible d'ajouter des données dans des base MongoDB sans s'y être connecté au prélable. Pour le faire, Il faut quitter le serveur.

```
exit
```

Les importations de données

Nous allons le faire grâce à l'outil `mongoimport`.

Importer des fichiers .json et .csv

Importer des données depuis un fichier JSON

```
mongoimport --jsonArray --db dev --collection collection_name --file movies.json
```

TAF: Commentez le résultat de cette commande. A quel élément d'un modèle relationnel vous fais penser cer résultat?
TAF: Vérifier que les données sont bien importées au bon endroit.
Importer des données depuis un fichier CSV

```
mongoimport --type csv -d productDB -c products --headerline --drop products.csv --uri
mongodb+srv://brice:bYUwmeLyq8yEW@cluster0.rdty5gv.mongodb.net
```

TAF: Vérifier que les données sont bien importées au bon endroit.
Exercice: Manipulation des données importées.

- Combien de produits avons nous en stock?
- Afficher la liste de tous ces produits.
- Afficher en priorité les produit avec le stock le plus faible.
- Quel est le produit le plus cher?

Bonus: Afficher uniquement le(s) produit(s) qui sont à commander aux forunisseurs(étant donné que le stock minimal recommandé est 5 par produit).
N.B: La méthode insertOne permet d'insérer un document. Lorsque le champ `_id` n'est pas mentionné, mongoDB va créer ce champ et lui affecter une valeur, dans ce cas de type « Object ID ». L'unicité est garantie.
Même s'il est possible de stocker dans un même champ des données de type différent, ce n'est pas une bonne pratique.

Cloud Database

Load sample datasets

Load sample Dataset

```
mongosh mongodb+srv://{USERNAME}:{PASSWORD}@cluster0.rdty5gv.mongodb.net
```

TAF: Vérifier que l'existence des données chargées. Il s'agit de 9 databases nommées suivant le pattern sample_

Exploration de données avec Mongo Compass

Généralités

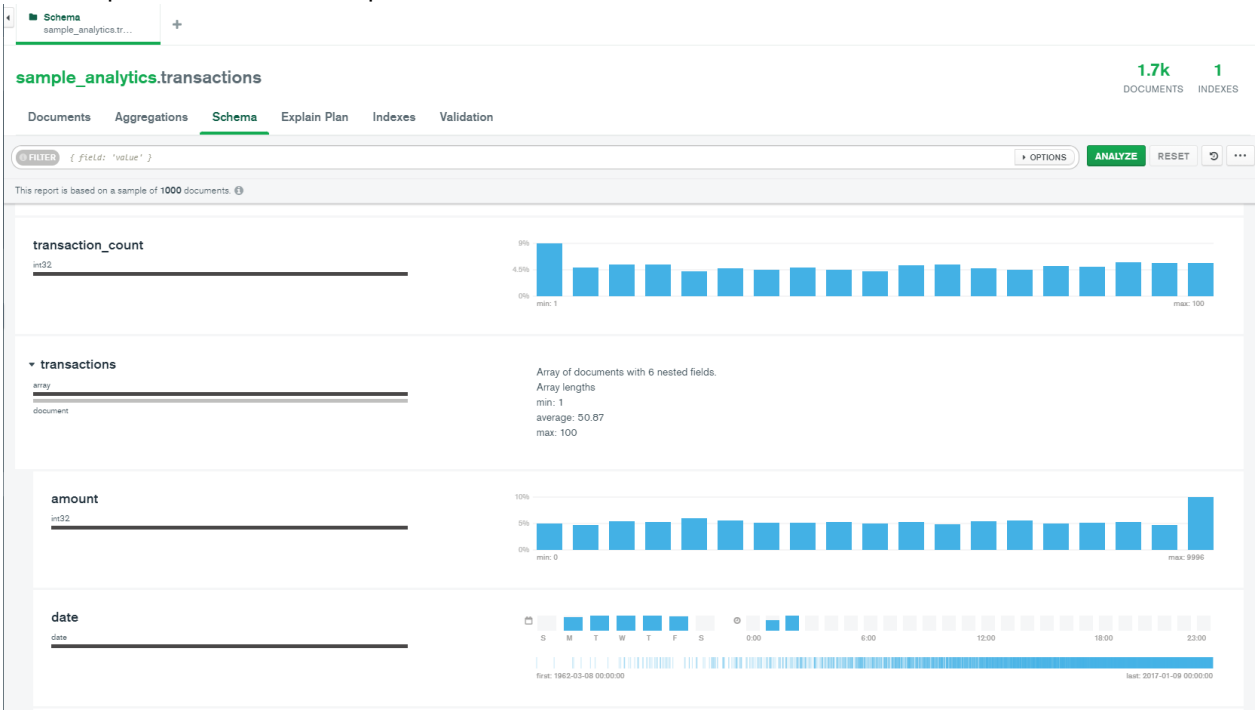
Se connecter au cluster Atlas sur Mongo Compass via la chaine de connexion du cluster.
Explorer la base de données sample_analystics.

TAF: Décrirer cette base de données en la comparant à une base de données relationnelle.

- Combien de tables y a t-il?
- Y a t il des relations entre les tables? Si oui Lesquelles?
- La notion de normalisation est elle respectée dans cette base de données? Expliquer.

TAF: Décrire les données d'un document de la collection `customers`.

- Quels sont les types de chaque champ?
- Ces types sont-ils tous utilisables dans une base de données type MySQL? Sinon, lesquels?
  Analyse de schéma
  Analyser le schéma de la collection `transactions` en allant vers l'onglet **Schema**. Cliquer sur **Analyze schema** pour avoir une description détaillée de la collection.



  Les indicateurs et graphiques affichés sont collection sur un échantillon réduit des données de la collection.
  TAF: Quel symbole intervient le plus dans les transactions?
  TAF: Combien y-a til de codes de transactions? Lesquels?
  TAF: A quelle heure y a il le plus de transcations?

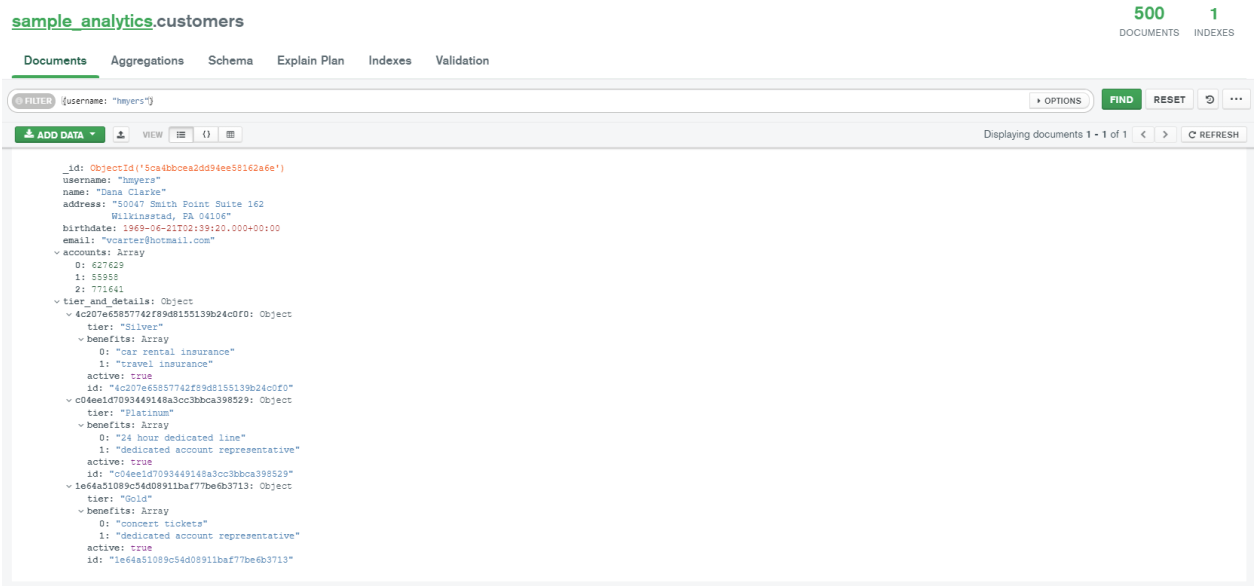
Requêtes dans Mongo Compass

Pour effectuer les requêtes, retourner dans l'onglet **Documents**.
Cas d'usage:

- Je suis agent d'un établissement financier et je souhaite exploiter des données de transactions de nos clients afin de leur recommander des produits adaptés. J'ai identifié un utilisateur type qui a réagit favorablement à mes recommandations. Je souhaite comprendre son profil et identifier des utilisateurs similaires.

Requêtes sur des champs simples

- Retrouver l'utilisateur et les informations disponibles sur lui/elle.



> Il s'agit visiblement d'une femme née en Juin 1969, vivant dans le Wilkinsstad et possédant 3 comptes.

Comme affiché sur l'image ci-dessus,
Le **filtre Compass** est: `{username: "hmyers"}`
L'équivalent de cette requête sur **Mongo Shell** est:

```
sample_analytics>db.customers.find({username: "hmyers"})
```

TAF: Retrouver les informations sur son premier compte (627629)
TAF: Quels produits détient cet utilisateur dans son compte?
TAF: Combien de transactions a réalisé ce compte?

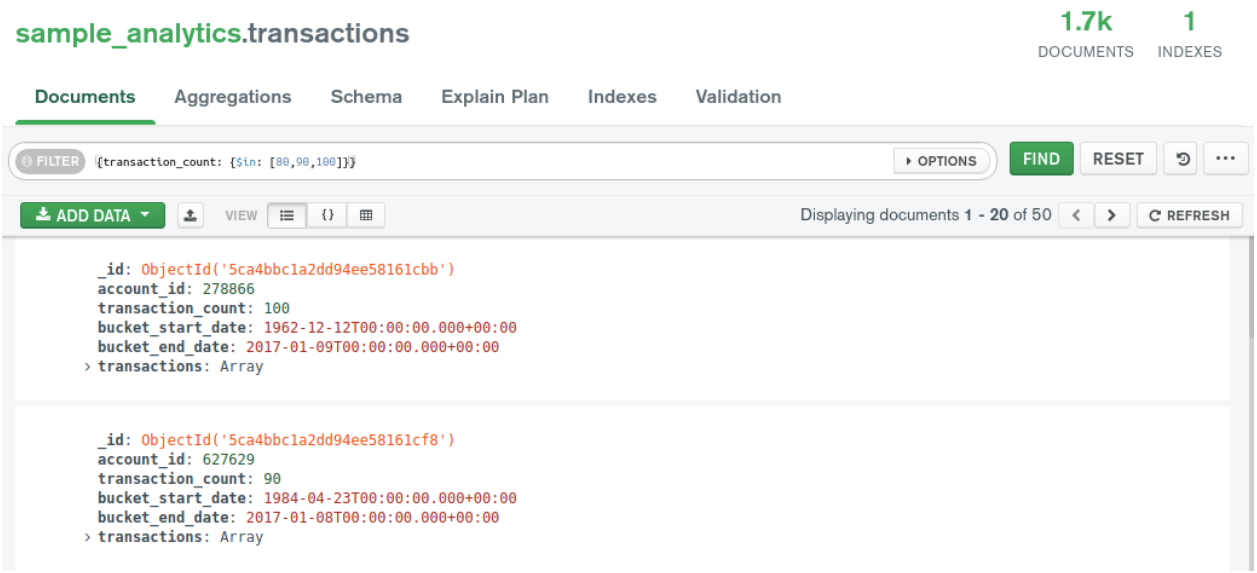
Nous pourrons faire une analyse plus détaillée par la suite.

Les opérateurs de comparaison

Documentation : https://docs.mongodb.com/manual/reference/operator/query-comparison/

- J'aimerais savoir quels comptes effectuent autant de transaction que **hmyers** avec une marge de +/- 10 transactions. Ce sont peut-être des utilisateurs avec un profil similaire. Pour l'instant, je ne veux

que les comptes avec 80,90 ou 100 transactions.



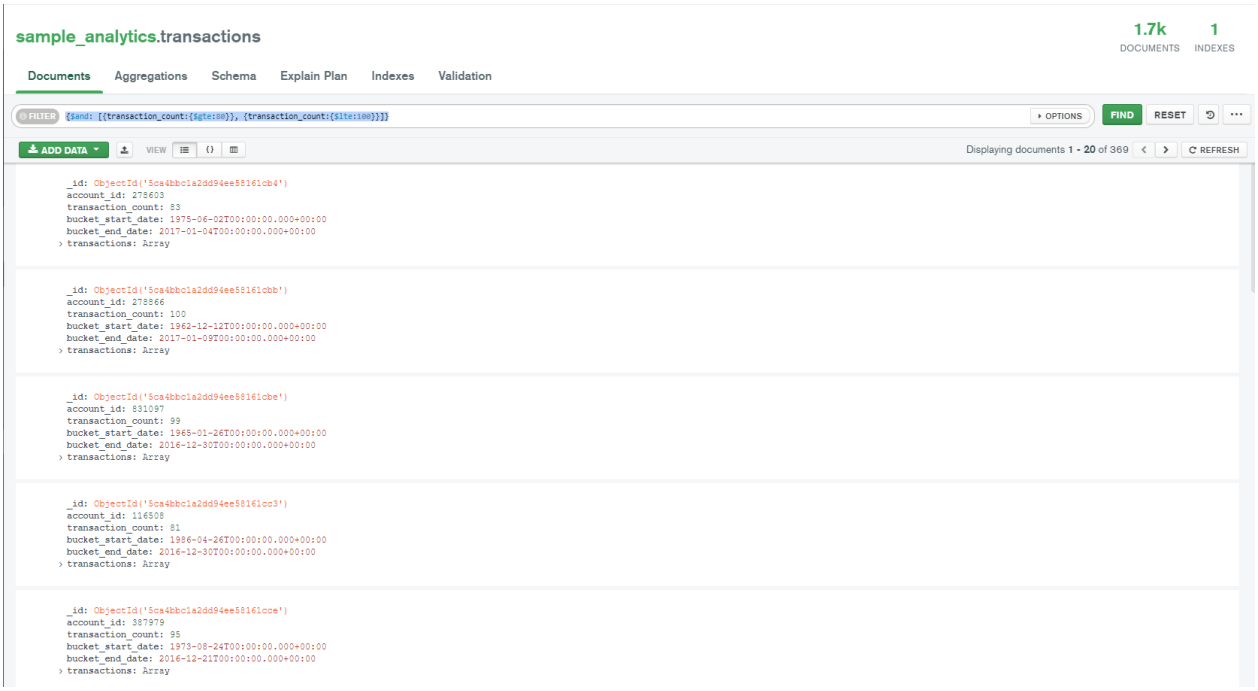
- Filtre Compass: `{transaction_count: {$in: [80,90,100]}}`

- **TAF: Rédiger la requête *Mongo Shell* qui retourne le nombre de comptes respectant cette condition.**

Les opérateurs logiques

Documentation : https://docs.mongodb.com/manual/reference/operator/query-logical/

- Cette estimation n'est pas vraiment fine, j'aimerai savoir précisément combien de comptes font entre 80 et 100 transactions.



- Filtre Compass: `{$and: [{transaction_count:{$gte:80}}, {transaction_count:{$lte:100}}]}`
- **TAF: Rédiger la requête *Mongo Shell* qui retourne le nombre de comptes respectant ayant effectué entre 80 et 100 transactions.**

Tri des résultats

- Pour faciliter ma prise ma campagne de recommandations, je dois prioriser les prises de contacts, ce qui revient à trier les comptes selon des critères. Mes 2 premiers critères sont le nombre de transactions et l'ancienneté des comptes.
  Je souhaite contacter en priorité les comptes les plus actifs et parmis celà, démarrer par les plus anciens.
- **Tri Compass**: `{transaction_count:-1, bucket_start_date:1}`
  -1 spécifie un ordre **décroissant**
  1 spécifie un ordre **croissant**
- **TAF: Rédiger la requête *Mongo Shell* qui retourne les comptes ayant entre 80 et 100 trasactions, en respectant les critères de priorité définis.**

Recherche dans des documents imbriqués

- Un autre paramètre déterminant dans mes choix de comptes est l'indice qu'achètent ou vendent mes comptes cibles. Pour optimiser mes chances de réussite, je choisis de contacter des personnes qui ont investi dans des boites au moins une boite tech notamment **Google**.
- Filtre Compass: `{"transactions.symbol":"goog"}`
  TAF: Rédiger la requête *Mongo Shell* qui retourne dans l'ordre souahité, les comptes respectant tous les critères précédant et qui en plus ont des actifs Google
  TAF: Ce critère a t il réduit la liste de comptes à cibler? Si oui, de combien? (requête à l'appui)

Recherche sur des tableaux (Array)

Maintenant que j'ai une liste de comptes restreinte selon les transactions, je souhaite filtrer les comptes selons leurs contenu en terme de produits.

- Je souhaite exclure les comptes:
  - Non diversifiés: La requête à taper dans Compass est: `{products: {$not : {$size: 1}}}`
  - Comportant des produits dérivés: La requête à taper dans Compass est: `{products: {$ne: "Derivatives"}}`
- Je souhaite avoir la liste des comptes qui ont le produit **Brokerage** en premier : `{"products.0": "Brokerage"}`
- J'aimerais aussi savoir combien de comptes ont le produit : **investmentFund**: `{"products": "InvestmentFund"}`

Projections

Les projections permettent de limiter les champs à afficher.
Lorsque je cherche les comptes qui ont le produit **investmentFund**, l'information finale qui m'intéresse estl'identifiant unique de chaque compte.
Pour avoir ce résultat, j'applique une projection à ma requête `{"products": "InvestmentFund"}`

- **Projection Compass**: `{_id:0, account_id:1}`
  Le champ `_id` est inclus par défaut dans le résultat. Dans la requête de projection `{_id:0, account_id:1}` ci-dessous, le champ `account_id` est retenu, le champ `_id` est exclu.

- **Projection Mongo Shell**

```
sample_analytics>db.accounts.find(
        {"products":  "InvestmentFund"},
        {_id:0, account_id:1}
)
[
  { account_id: 383777 }, { account_id: 794875 },
  { account_id: 260499 }, { account_id: 299072 },
  { account_id: 977982 }, { account_id: 212024 },
  { account_id: 433811 }, { account_id: 403363 },
  { account_id: 276528 }, { account_id: 383701 },
  { account_id: 463155 }, { account_id: 895735 },
  { account_id: 984021 }, { account_id: 503933 },
  { account_id: 475387 }, { account_id: 775690 },
  { account_id: 136137 }, { account_id: 522933 },
  { account_id: 785786 }, { account_id: 462501 }
]
```

Update

Par appel téléphonique, la cliente portant le nom "Katherine David" m'a indiqué un changement d'adresse e-mail..
Sa nouvelle adresse est `katherine.david@gmail.com`.

```
sample_analytics>db.customers.updateOne(
{  name: "Katherine David"},
{ $set: { email: "katherine.david@gmail.com"}}
)
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Delete

Le service juridique m'informe de la volonté d'un client de supprimer ses données client. Ils'agit de Brad Cardenas.

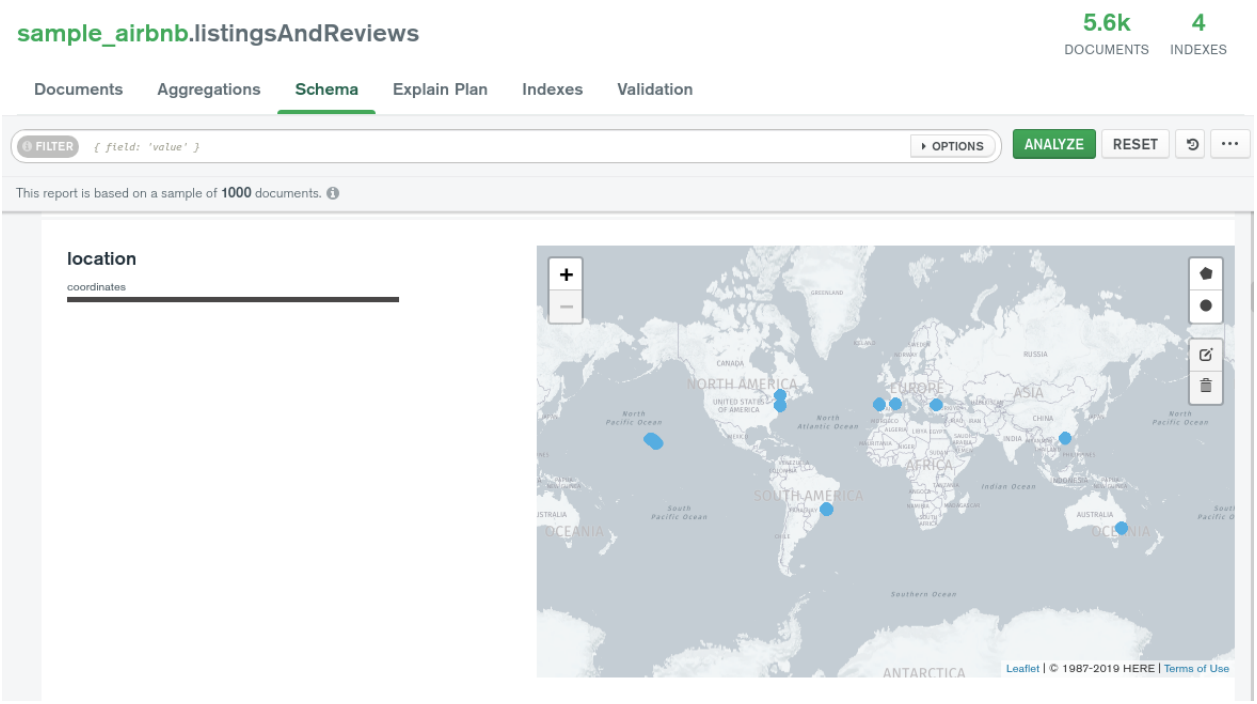
```
sample_analytics>db.customers.deleteOne({"name": "Brad Cardenas"})
{ acknowledged: true, deletedCount: 1 }
```

Rechercher géospatiale

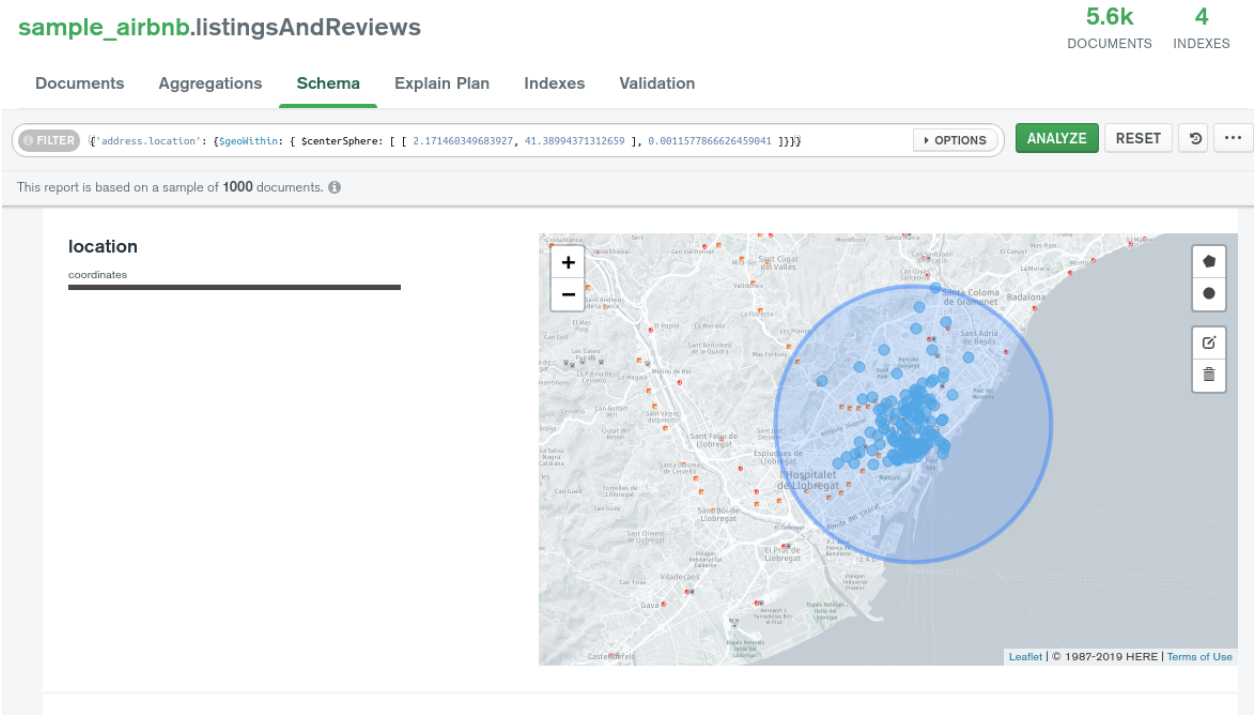
Documentation: https://www.mongodb.com/docs/manual/reference/operator/query-geospatial/
Cas d'usage : AirBNB

- J'ai accès à la base de données de AirBNB, je souhaite trouver des logements au tour de barcelone avec un minimum de 3 commentaires sans dépôt de garantie
- `{'address.location': {$geoWithin: { $centerSphere: [ [ 2.1723029405703502, 41.401529208292374 ], 0.0017309972845589124 ]}}}`
  Sur MongoDB Compass:

- Accéder à la collection `sample_airbnb.listingsAndReviews`
- Aller dans l'onglet **Schema**
- Consulter le champ `address.location`
- Au niveau de la Map, zoomer sur la ville de Barcelone
- Cliquer sur le symbole **Draw a circle**(⚫)



- Puis dessiner un cercle au tour de la zone d'intérêt



La requête qui apparaît dans le champ filter est celle qui permet de restreindre les recherches à une zone géographique précise.
Analysons cette requête

```
{'address.location':
                {$geoWithin:
                        { $centerSphere: [
                                [ 2.171460349683927, 41.38994371312659 ],
                                0.0011577866626459041
```

```
                                                    ]
                        }
                }
})
```

- address.location est le champ cible
- $geoWithin est l'opérateur de sélection géospatial qui permet de sélectionner des géomtries dans un espace défini
- $centerSphere est un spécificateur de surface géométrique. Ce dernier spécifie un cercle de centre `(2.171460349683927, 41.38994371312659)` et rayon `0.0011577866626459041`
  TAF: Rédiger la requête *Mongo Shell* permettant de lister les proches selon ces critères:
- Proche de Rio de Janeiro
- Avec 2 lits maximum
- Nuitée inférieure à 100€
- Propriété de type appartement
- Avec WiFi et Cuisine

Recherche de documents ayant des champs non renseignés

Pour mon voyage, afin d'avoir une minimum d'assurance, je ne veux que des logments avec des commentaires. Lire les avis et commentaires me permettront de choisir sereinement.
Retrouver les documents dont les champs sont nn renseignés peut se faire de 3 manière:
1- Pour retourner les documents pour lesquels le champ `first_review` n'existe pas:

- **Filtre Compass:** `{"first_review": {$exists: false}}`
  2- Pour retourner les documents pour lesquels le champ `first_review` vaut null ainsi que ceux pour lesquels le champ n'existe pas
- **Filtre Compass:** `{"first_review": null }`
  3- Pour récupérer uniquement ceux dont le champ vaut null, on teste par rapport au type du champ, `null`
  correspondant au **type 10 (BSON type)**.
- **Filtre Compass:** `{"first_review": {$type: 10} }`
  TAF: Rédiger la requête Mongo Shell, permettant de lister les logements respectant les 5 critères précédents avec en plus une restriction sur ceux qui ont au moins 1 commentaire. Je souhaite voir en priorité les logments les moins chers.
  TAF: Comparer le nombre de résultats avec la requête précédente pour vérifier l'impact.