Mongo

show dbs

use shop

db.products.insertOne({name:"Book", price : 9.99})

db.products.find() db.products.find().pretty()

db.products.updateOne({name:"iPhone"},{$set:{ price : 1119.99}})

db.products.find({price {$gt :4}})

db.products.insertOne({name:"iPhone XS", price : 19.99, details:{ram:"3gb", battery:"2500mAh"}})

db.products.find({"details.ram":"3gb"}).pretty()

db.products.insertOne({name:"iPhone 11", price : 1200.99, details:{ram:"4gb",
battery: »2500mAh"},tags:["mobile","apple","toto"]})

db.products.updateOne({name:"iPhone 11 » },{$addToSet:{tags:"ordi"}})

db.products.updateOne({tags:"ok"},{$set:{tags.$:"FRANCE"}})

Expliquer en quoi ça consiste

Indexes :
Aggregates : projection —>

mongod -h

config file pour voir les docs

mongoDBCompass : GUI

crud : Create, Read, Update, Delete

```
db.products.find().pretty()
```

```
db.products.insertMany([{ _id: "Article", name:
« chasse et peche", date: 10/03, auteur: Pablo},
{ _id: "Article2", name: « Macron Démission", auteur:
Brigitte, date: 6/01 }, { _id: « article3", name:
"Téléfoot", date: 08/09, auteur: Denis Brognard}])
```

```
db.products.insertMany([{ _id: "Citroen", name:
"Citroen", prix: 50000 }, { _id: "AMG", name: "AMG",
prix: 95000 }, { _id: "Aston", name: "Aston", prix:
89000 }], { ordered: false })
```

Rajoute les nouvelles données

```
db.products.insertOne({ name: "object" }, { w: 0 } )
```

Read :
2 types d'opérateurs
- Query Operators - trouver data

condition logique (comparaison)
geospacial
- Projection Operators - modifier l'affichage

Utiliser les opérateurs LTE et GT

db.user.find({ age: { $lte: 20 } }).pretty()

db.user.find({ age: { $gt: 20 } }).pretty()

Trouver tous les array

Opérateurs IN et NIN (not in)

db.user.find({ age: { $in: [5, 15] } }).pretty()

db.user.find({ age: { $nin: [5, 15] } }).pretty()

Opérateur OR

db.user.find({ $or: [{ age: { $lt: 20 } }, { price: 10 }] })

Opérateur AND

db.user.find({ $and: [{ age: { $ne: 18 } }, { index: {$in: [2, 4, 6, 8]} }] }).pretty()

Opérateur X un certain champs avec un certain type.
Si le document à une certaine propriété

db.users.find({friends: {sin: [{id: 0, name: « Matilda Schultz » }]}}).pretty()

db.users.find({tags: {$exists: false}}).pretty()

```
> db.users.find({tags: {$type: « Array"}}).pretty()
```

Validation Action :
- erreur
- warning

Validation level :
- insert
- insert and update

```
 db.createCollection("inventaires", { validator:
{ $jsonSchema: { required: [ 'name' ] } }  })
```

```
db.inventaires.insertOne({ prix: 55 })
```

Créer une BDD
Schéma de la conception
Utilisateurs qui se connecte et qui poste des articles et qui commente
D'autres user peuvent juste commenter

Article avec texte, date, auteurs
espace commentaire

voir son nom, la date et le commentaire (collections)

A rendre :
- Schéma

use article
use commentaire
---
db.createUser({ user: "Admin44", pwd: "passw0rd", roles: [{ role: "readWrite", db: "article" }, { role: "readWrite", db:"commentaire" }] })

db.createUser({ user: "User44", pwd: "passw0rd", roles: [{ role: "read", db: "article" }, { role: "readWrite", db:"commentaire" }] })
---
db.article.insertOne({name:"chasser", details:{auteur:"patdeph", date:"23/03"}})

db.commentaire.insertOne({name:"bel article", details:{auteur:"jeff", date:"24/03"}})

/base de données communes
/créer collection article et commentaire dans cette base
/embed document car ils commentaire et article fortement lié

```
db.devoir.insert(
{
   "article":"Honkytonk Man",
   "details":{
      "nom":"Eastwood",
      « date »:"23/02"
   },

"commentaire":"Honkytonk Man",
   "details":{
      "nom":"jeff",
      « date":"24/02"

}
)
```