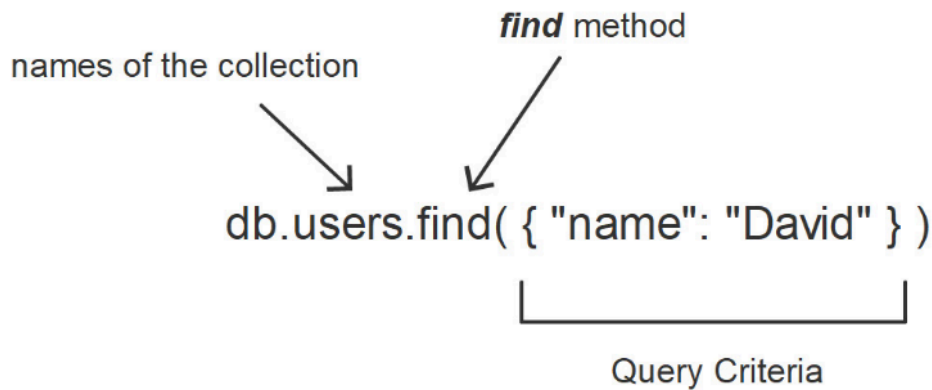


MONGODB



```
// Select d'un champ spécifique
db.unicorns.find({}, {name: 1});
```

```
Je ne veux pas l'id
db.unicorns.find({}, {name: 1, _id:0 });
```

```
// classer les licornes les plus lourdes d'abord
db.unicorns.find().sort({weight: -1})
```

La pagination des résultats peut être réalisée par `limit` et de saut de curseur (`skip`).
Exemple : pour obtenir les deuxième et troisième licornes les plus lourdes

```
db.unicorns.find()
  .sort({weight: -1})
  .limit(2)
  .skip(1)
```

Count

```
db.unicorns.count({ vampires: {$gt: 50}})
```

// Modification multiples

```
db.unicorns.update({ },
  {$set: {vaccinated: true }},
  {multi:true});

db.unicorns.find({vaccinated: true});
```

```
db.animaux.updateMany({},
  {$set: {weight: 200 }},
  );
```

je recherche tous les attributs ayant le type food mais sans afficher le type.

```
db.inventory.find( { type: 'food' }, { type:0 } )
```

NO JOIN

ici on cree un employe "manager"

```
db.employees.insert({_id: ObjectId("4d85c7039ab0fd70a117d730"),
                      name: 'Leto'})
```

```
db.employees.insert({_id: ObjectId(
    "4d85c7039ab0fd70a117d731"),
    name: 'Duncan',
    manager: ObjectId("4d85c7039ab0fd70a117d730")});
```

```
db.employees.insert({_id: ObjectId(
    "4d85c7039ab0fd70a117d732"),
    name: 'Moneo',
    manager: ObjectId("4d85c7039ab0fd70a117d730")});
```

Un employé a plusieurs managers

```
db.employees.insert({_id: ObjectId(
    "4d85c7039ab0fd70a117d733"),
    name: 'Siona',

    manager: [      ObjectId(
        "4d85c7039ab0fd70a117d730"),
        ObjectId(
        "4d85c7039ab0fd70a117d732")]
})
```