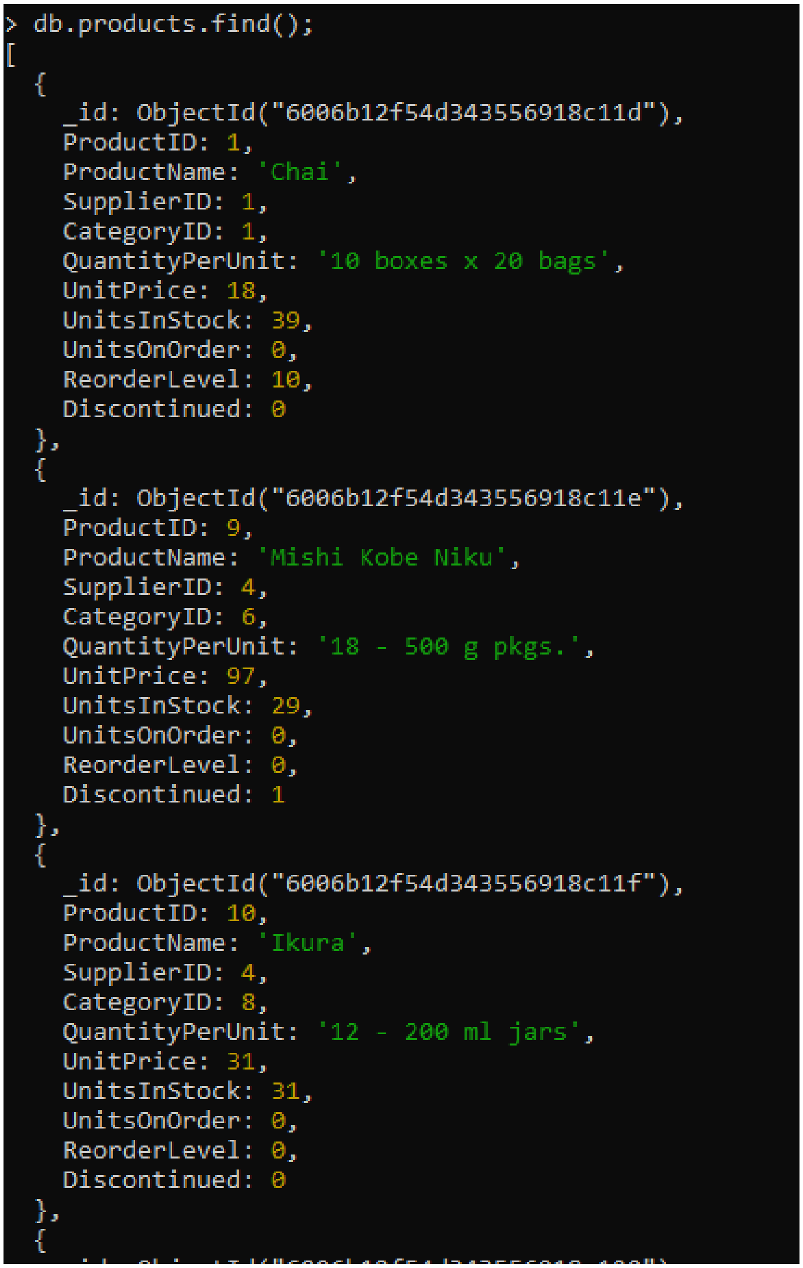
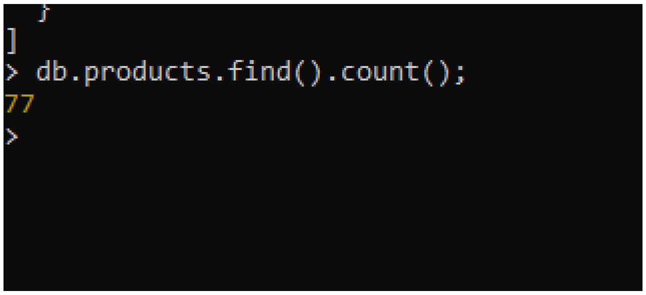
Création de base de données à base d'un fichier

db.products.find()



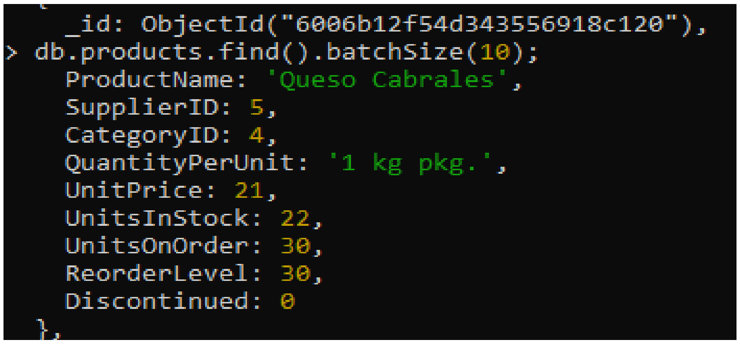
```
> db.products.find();
[
  {
    _id: ObjectId("6006b12f54d343556918c11d"),
    ProductID: 1,
    ProductName: 'Chai',
    SupplierID: 1,
    CategoryID: 1,
    QuantityPerUnit: '10 boxes x 20 bags',
    UnitPrice: 18,
    UnitsInStock: 39,
    UnitsOnOrder: 0,
    ReorderLevel: 10,
    Discontinued: 0
  },
  {
    _id: ObjectId("6006b12f54d343556918c11e"),
    ProductID: 9,
    ProductName: 'Mishi Kobe Niku',
    SupplierID: 4,
    CategoryID: 6,
    QuantityPerUnit: '18 - 500 g pkgs.',
    UnitPrice: 97,
    UnitsInStock: 29,
    UnitsOnOrder: 0,
    ReorderLevel: 0,
    Discontinued: 1
  },
  {
    _id: ObjectId("6006b12f54d343556918c11f"),
    ProductID: 10,
    ProductName: 'Ikura',
    SupplierID: 4,
    CategoryID: 8,
    QuantityPerUnit: '12 - 200 ml jars',
    UnitPrice: 31,
    UnitsInStock: 31,
    UnitsOnOrder: 0,
    ReorderLevel: 0,
    Discontinued: 0
  },
  {
```

Les documents sont inclus dans la collection « products »



```
  }
]
> db.products.find().count();
77
>
```

Modifier la taille du batch de MongoDB pour afficher juste 10 éléments.



```
    _id: ObjectId("6006b12f54d343556918c120"),
> db.products.find().batchSize(10);
    ProductName: 'Queso Cabrales',
    SupplierID: 5,
    CategoryID: 4,
    QuantityPerUnit: '1 kg pkg.',
    UnitPrice: 21,
    UnitsInStock: 22,
    UnitsOnOrder: 30,
    ReorderLevel: 30,
    Discontinued: 0
  },
```

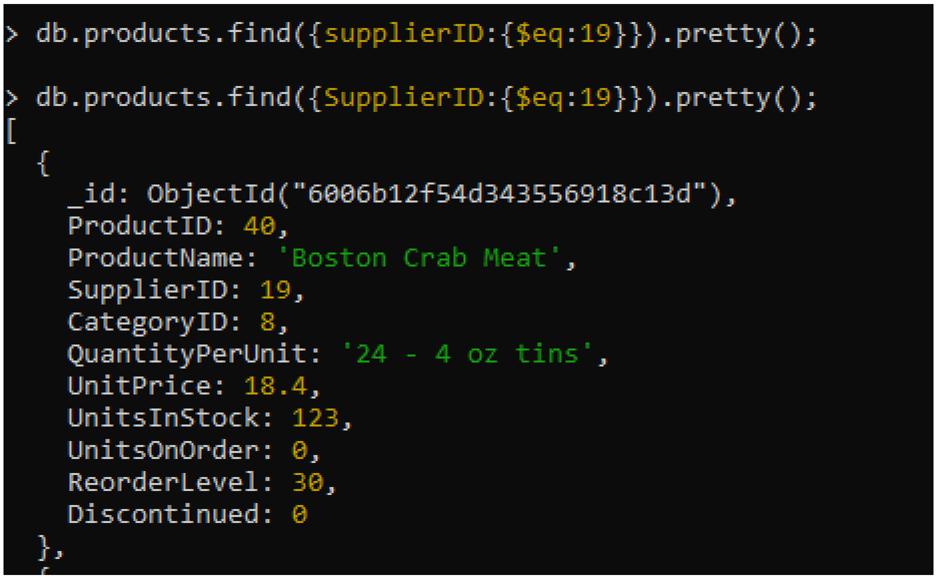
itère sur les 20 premiers documents

Exécuter les deux commandes suivantes l'une après l'autre

db.products.find({supplierID:{$eq:19}}).pretty();

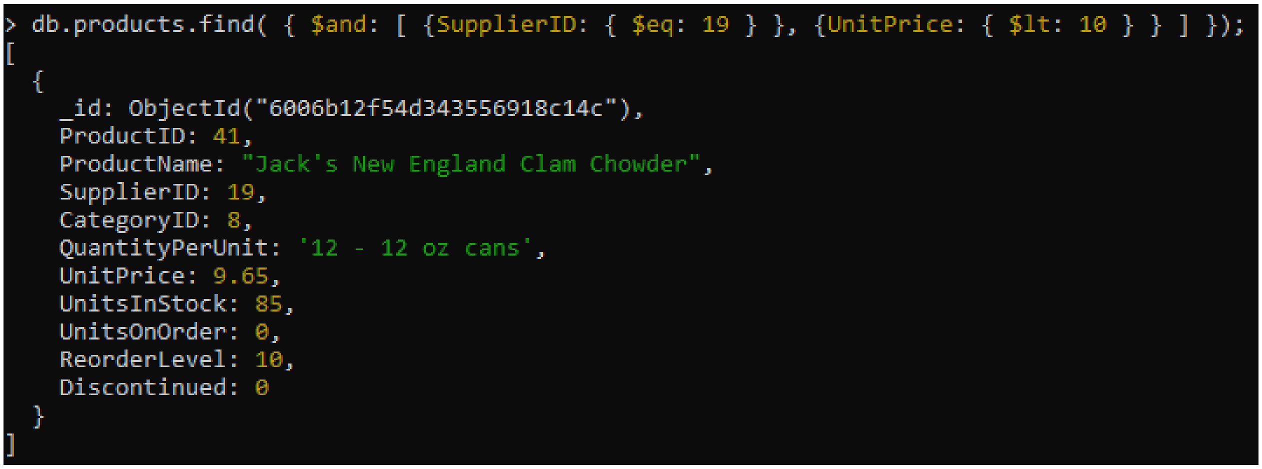
db.products.find({SupplierID:{$eq:19}}).pretty();

Remarque : la première ne retourne rien car il n'est existé pas, contre la deuxième il est existé (supplierID vs SupplierID)



```
> db.products.find({supplierID:{$eq:19}}).pretty();

> db.products.find({SupplierID:{$eq:19}}).pretty();
[
  {
    _id: ObjectId("6006b12f54d343556918c13d"),
    ProductID: 40,
    ProductName: 'Boston Crab Meat',
    SupplierID: 19,
    CategoryID: 8,
    QuantityPerUnit: '24 - 4 oz tins',
    UnitPrice: 18.4,
    UnitsInStock: 123,
    UnitsOnOrder: 0,
    ReorderLevel: 30,
    Discontinued: 0
  },
```

Sélectionner les documents qui ont un SupplierID égale à 19 et UnitPrice moins que 10



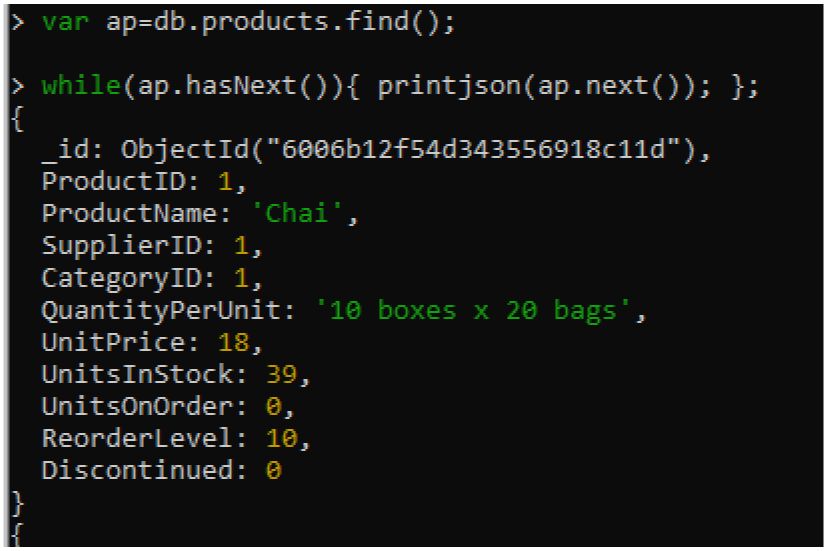
```
> db.products.find( { $and: [ {SupplierID: { $eq: 19 } }, {UnitPrice: { $lt: 10 } } ] });
[
  {
    _id: ObjectId("6006b12f54d343556918c14c"),
    ProductID: 41,
    ProductName: "Jack's New England Clam Chowder",
    SupplierID: 19,
    CategoryID: 8,
    QuantityPerUnit: '12 - 12 oz cans',
    UnitPrice: 9.65,
    UnitsInStock: 85,
    UnitsOnOrder: 0,
    ReorderLevel: 10,
    Discontinued: 0
  }
]
```

Dans le shell, créer la variable suivante :

var allProducts=db.products.find();

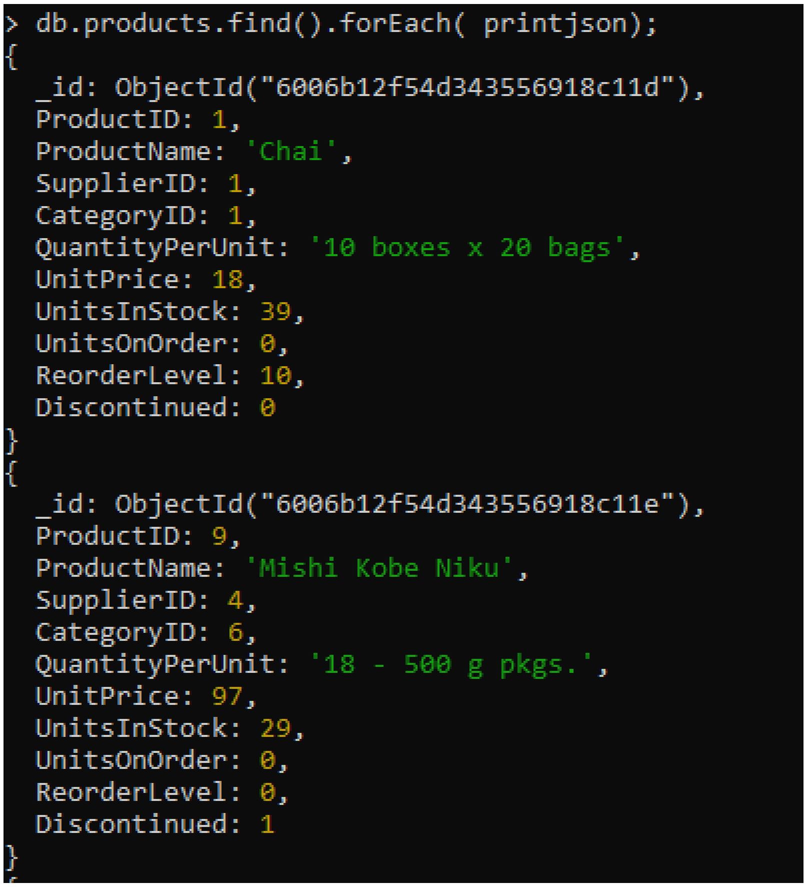
Exécuter le bout de code suivant dans votre shell

while(ap.hasNext()){ printjson(ap.next()); };



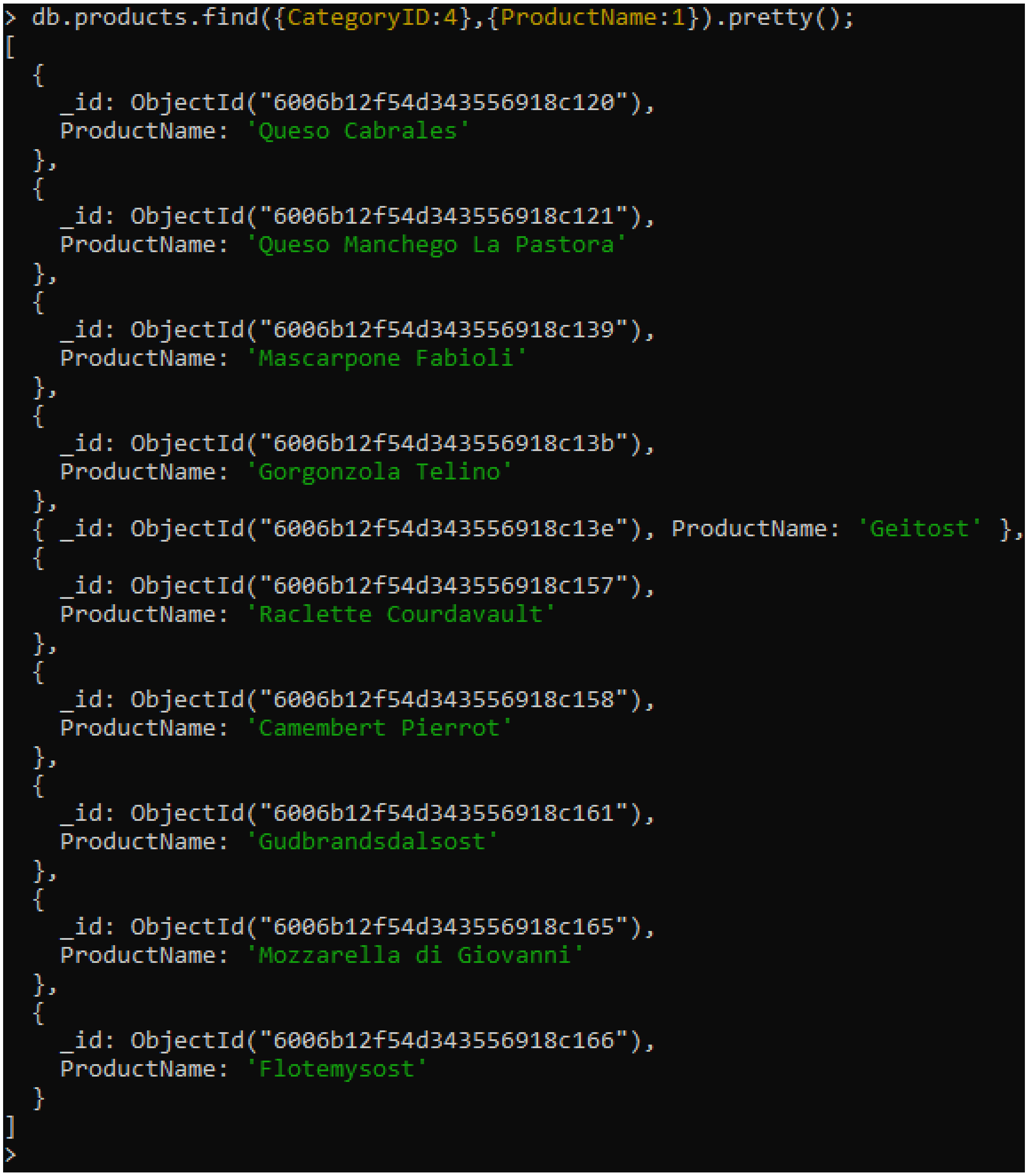
```
> var ap=db.products.find();

> while(ap.hasNext()){ printjson(ap.next()); };
{
  _id: ObjectId("6006b12f54d343556918c11d"),
  ProductID: 1,
  ProductName: 'Chai',
  SupplierID: 1,
  CategoryID: 1,
  QuantityPerUnit: '10 boxes x 20 bags',
  UnitPrice: 18,
  UnitsInStock: 39,
  UnitsOnOrder: 0,
  ReorderLevel: 10,
  Discontinued: 0
}
{
```

Afficher tous les éléments du curseur : db.products.find() en utilisant la fonction forEach.



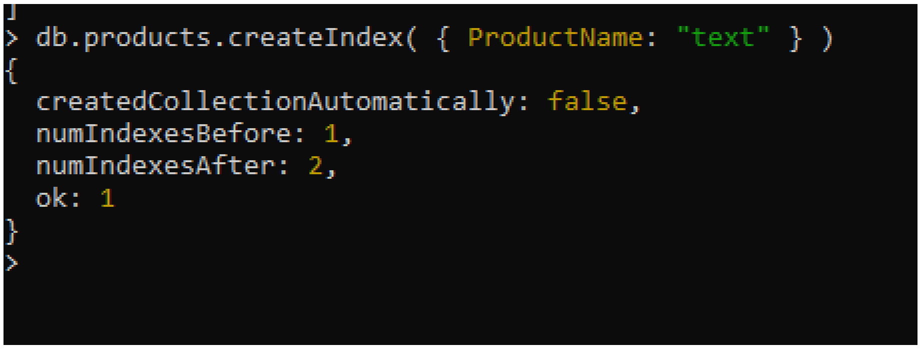
```
> db.products.find().forEach( printjson);
{
  _id: ObjectId("6006b12f54d343556918c11d"),
  ProductID: 1,
  ProductName: 'Chai',
  SupplierID: 1,
  CategoryID: 1,
  QuantityPerUnit: '10 boxes x 20 bags',
  UnitPrice: 18,
  UnitsInStock: 39,
  UnitsOnOrder: 0,
  ReorderLevel: 10,
  Discontinued: 0
}
{
  _id: ObjectId("6006b12f54d343556918c11e"),
  ProductID: 9,
  ProductName: 'Mishi Kobe Niku',
  SupplierID: 4,
  CategoryID: 6,
  QuantityPerUnit: '18 - 500 g pkgs.',
  UnitPrice: 97,
  UnitsInStock: 29,
  UnitsOnOrder: 0,
  ReorderLevel: 0,
  Discontinued: 1
}
```

db.products.find({CategoryID:4},{ProductName:1}).pretty();



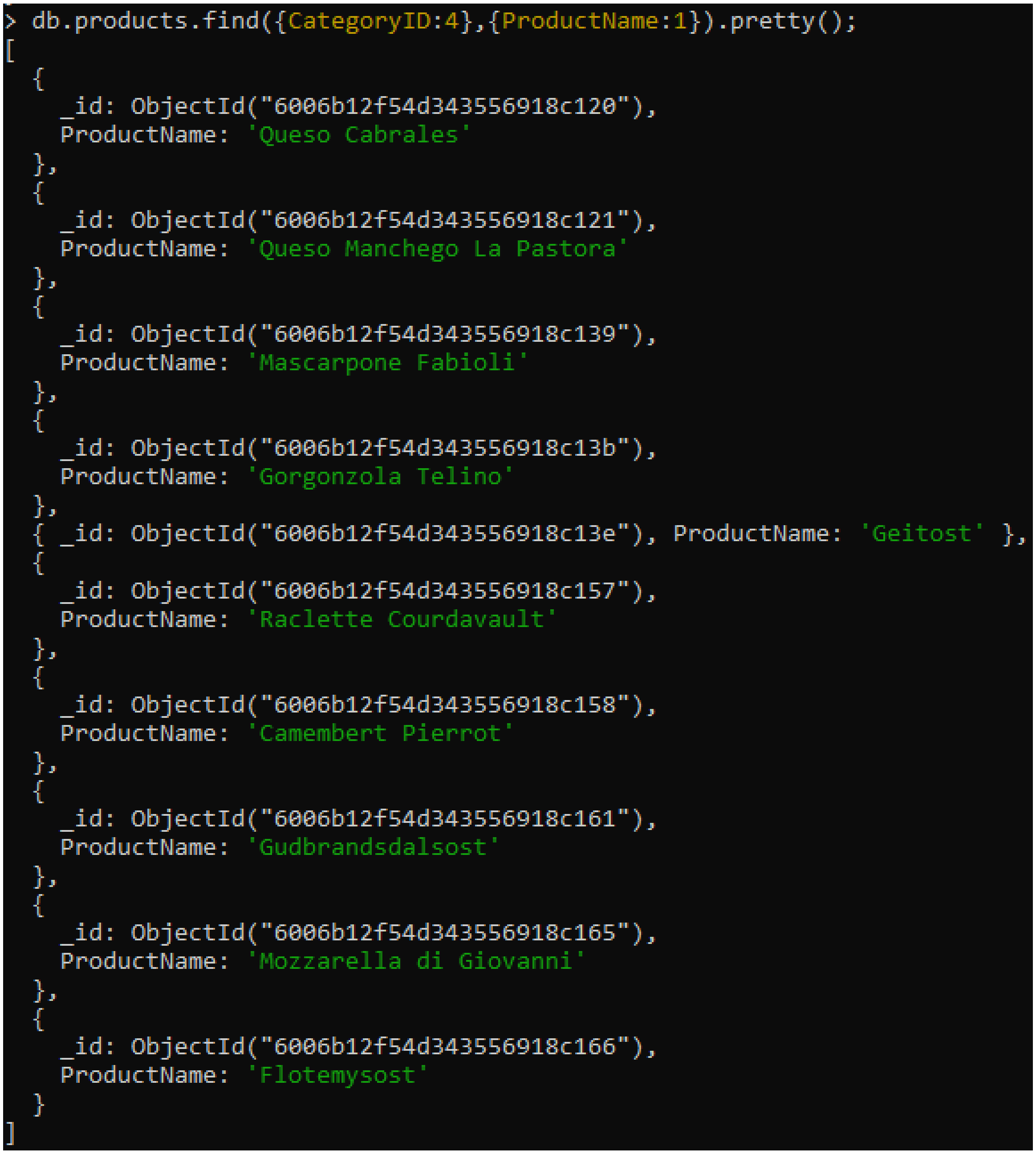
```
> db.products.find({CategoryID:4},{ProductName:1}).pretty();
[
  {
    _id: ObjectId("6006b12f54d343556918c120"),
    ProductName: 'Queso Cabrales'
  },
  {
    _id: ObjectId("6006b12f54d343556918c121"),
    ProductName: 'Queso Manchego La Pastora'
  },
  {
    _id: ObjectId("6006b12f54d343556918c139"),
    ProductName: 'Mascarpone Fabioli'
  },
  {
    _id: ObjectId("6006b12f54d343556918c13b"),
    ProductName: 'Gorgonzola Telino'
  },
  { _id: ObjectId("6006b12f54d343556918c13e"), ProductName: 'Geitost' },
  {
    _id: ObjectId("6006b12f54d343556918c157"),
    ProductName: 'Raclette Courdavault'
  },
  {
    _id: ObjectId("6006b12f54d343556918c158"),
    ProductName: 'Camembert Pierrot'
  },
  {
    _id: ObjectId("6006b12f54d343556918c161"),
    ProductName: 'Gudbrandsdalsost'
  },
  {
    _id: ObjectId("6006b12f54d343556918c165"),
    ProductName: 'Mozzarella di Giovanni'
  },
  {
    _id: ObjectId("6006b12f54d343556918c166"),
    ProductName: 'Flotemysost'
  }
]
>
```

En utilisant le shell, créer un index de type « text » pour « ProductName » :



```
]
> db.products.createIndex( { ProductName: "text" } )
{
  createdCollectionAutomatically: false,
  numIndexesBefore: 1,
  numIndexesAfter: 2,
  ok: 1
}
>
```

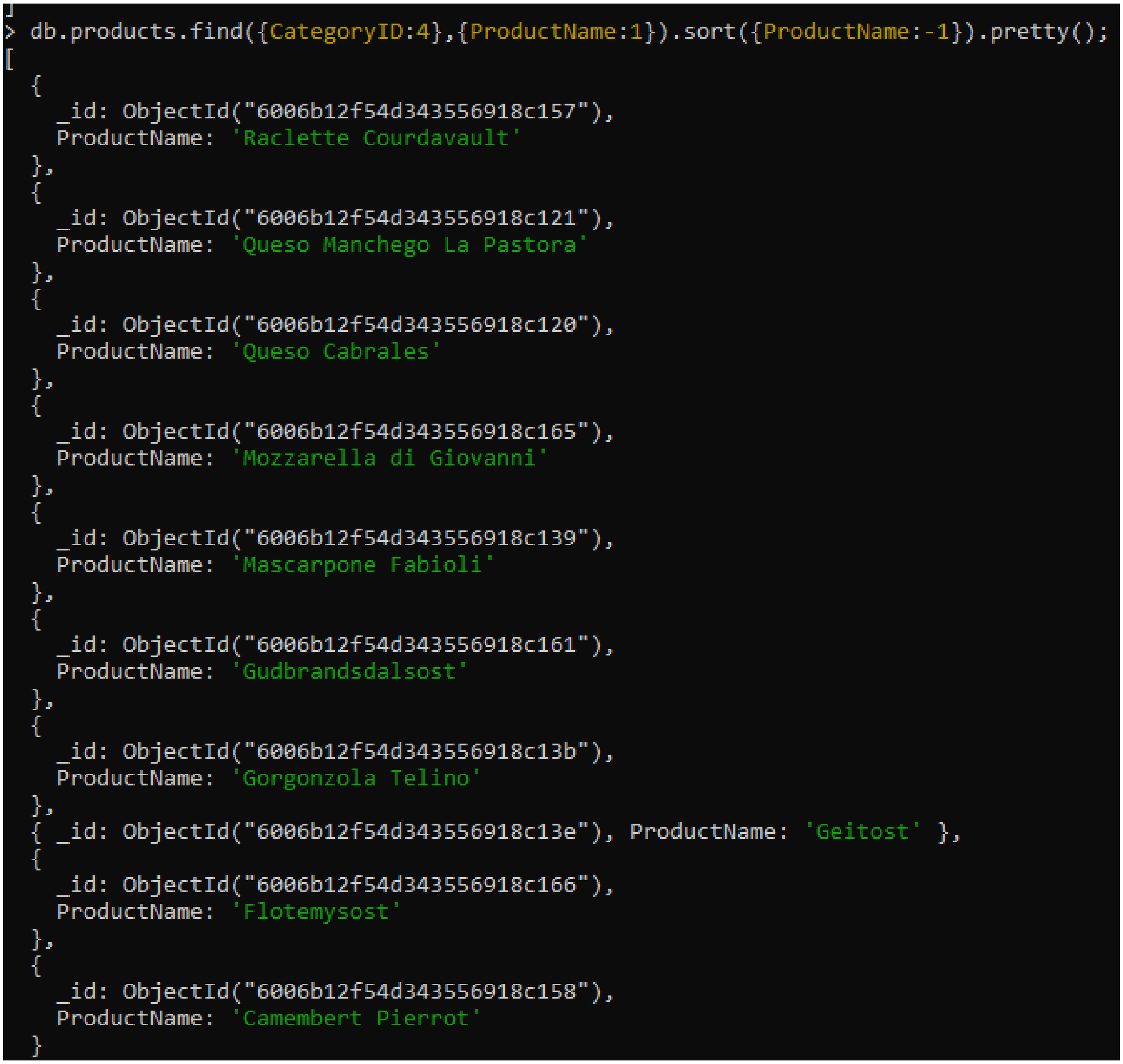
db.products.find({CategoryID:4},{ProductName:1}).pretty();



```
> db.products.find({CategoryID:4},{ProductName:1}).pretty();
[
  {
    _id: ObjectId("6006b12f54d343556918c120"),
    ProductName: 'Queso Cabrales'
  },
  {
    _id: ObjectId("6006b12f54d343556918c121"),
    ProductName: 'Queso Manchego La Pastora'
  },
  {
    _id: ObjectId("6006b12f54d343556918c139"),
    ProductName: 'Mascarpone Fabioli'
  },
  {
    _id: ObjectId("6006b12f54d343556918c13b"),
    ProductName: 'Gorgonzola Telino'
  },
  { _id: ObjectId("6006b12f54d343556918c13e"), ProductName: 'Geitost' },
  {
    _id: ObjectId("6006b12f54d343556918c157"),
    ProductName: 'Raclette Courdavault'
  },
  {
    _id: ObjectId("6006b12f54d343556918c158"),
    ProductName: 'Camembert Pierrot'
  },
  {
    _id: ObjectId("6006b12f54d343556918c161"),
    ProductName: 'Gudbrandsdalsost'
  },
  {
    _id: ObjectId("6006b12f54d343556918c165"),
    ProductName: 'Mozzarella di Giovanni'
  },
  {
    _id: ObjectId("6006b12f54d343556918c166"),
    ProductName: 'Flotemysost'
  }
]
```

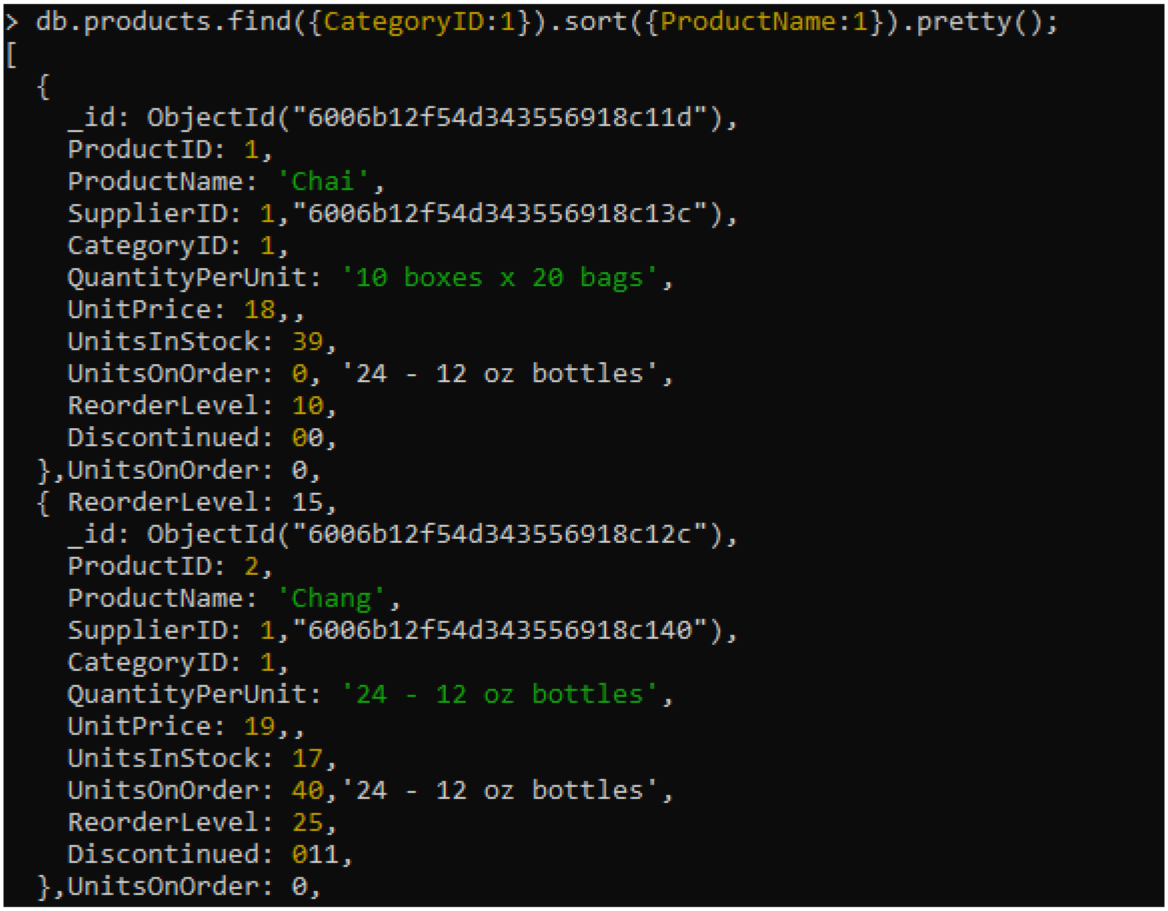
executer la commande suivante:

db.products.find({CategoryID:4},{ProductName:1}).sort({ProductName:-1}).pretty();



```
]
> db.products.find({CategoryID:4},{ProductName:1}).sort({ProductName:-1}).pretty();
[
  {
    _id: ObjectId("6006b12f54d343556918c157"),
    ProductName: 'Raclette Courdavault'
  },
  {
    _id: ObjectId("6006b12f54d343556918c121"),
    ProductName: 'Queso Manchego La Pastora'
  },
  {
    _id: ObjectId("6006b12f54d343556918c120"),
    ProductName: 'Queso Cabrales'
  },
  {
    _id: ObjectId("6006b12f54d343556918c165"),
    ProductName: 'Mozzarella di Giovanni'
  },
  {
    _id: ObjectId("6006b12f54d343556918c139"),
    ProductName: 'Mascarpone Fabioli'
  },
  {
    _id: ObjectId("6006b12f54d343556918c161"),
    ProductName: 'Gudbrandsdalsost'
  },
  {
    _id: ObjectId("6006b12f54d343556918c13b"),
    ProductName: 'Gorgonzola Telino'
  },
  { _id: ObjectId("6006b12f54d343556918c13e"), ProductName: 'Geitost' },
  {
    _id: ObjectId("6006b12f54d343556918c166"),
    ProductName: 'Flotemysost'
  },
  {
    _id: ObjectId("6006b12f54d343556918c158"),
    ProductName: 'Camembert Pierrot'
  }
```

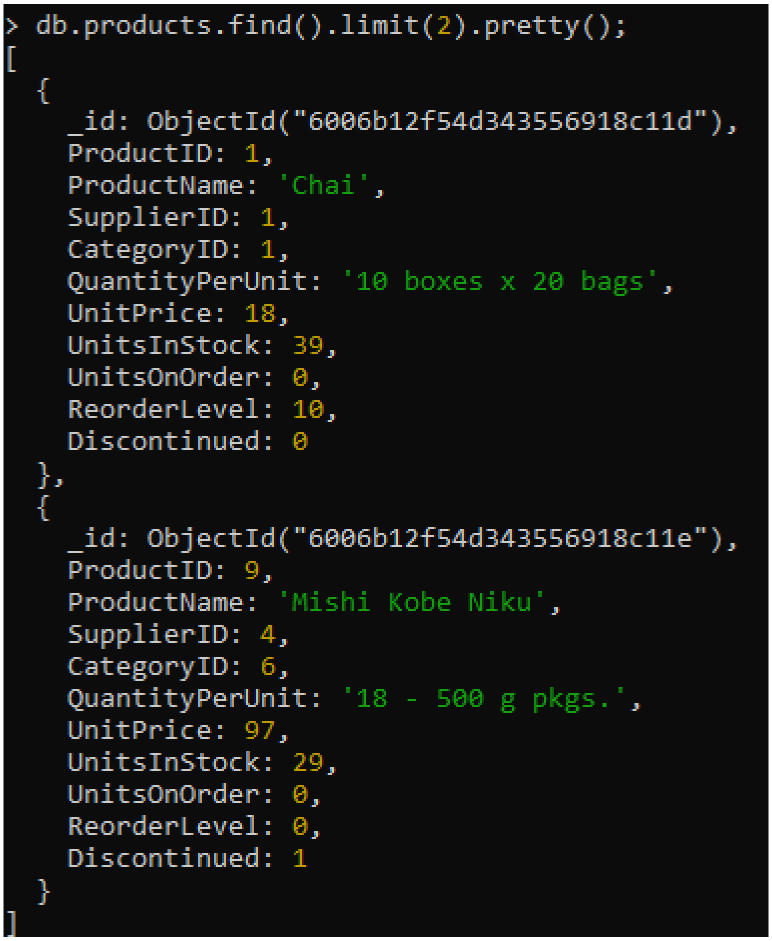
Afficher les documents de products ayant un « CategoryID » égal à un, par ordre ascendant de ProductName



```
> db.products.find({CategoryID:1}).sort({ProductName:1}).pretty();
[
  {
    _id: ObjectId("6006b12f54d343556918c11d"),
    ProductID: 1,
    ProductName: 'Chai',
    SupplierID: 1,"6006b12f54d343556918c13c"),
    CategoryID: 1,
    QuantityPerUnit: '10 boxes x 20 bags',
    UnitPrice: 18,,
    UnitsInStock: 39,
    UnitsOnOrder: 0, '24 - 12 oz bottles',
    ReorderLevel: 10,
    Discontinued: 00,
  },UnitsOnOrder: 0,
  { ReorderLevel: 15,
    _id: ObjectId("6006b12f54d343556918c12c"),
    ProductID: 2,
    ProductName: 'Chang',
    SupplierID: 1,"6006b12f54d343556918c140"),
    CategoryID: 1,
    QuantityPerUnit: '24 - 12 oz bottles',
    UnitPrice: 19,,
    UnitsInStock: 17,
    UnitsOnOrder: 40,'24 - 12 oz bottles',
    ReorderLevel: 25,
    Discontinued: 011,
  },UnitsOnOrder: 0,
```

Limiter le nombre des documents à afficher à 2 ;

db.products.find().sort({ProductID:-1}).limit(1).pretty()

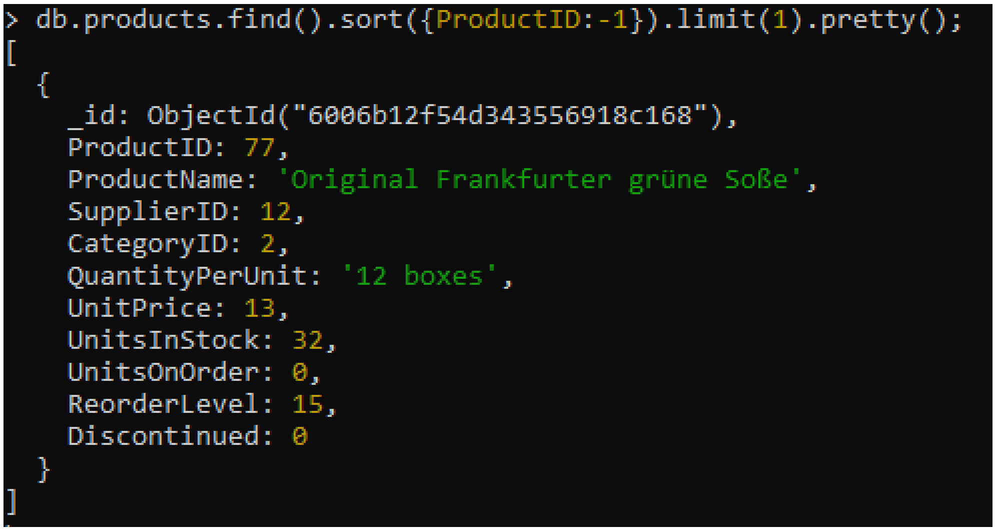


```
> db.products.find().limit(2).pretty();
[
  {
    _id: ObjectId("6006b12f54d343556918c11d"),
    ProductID: 1,
    ProductName: 'Chai',
    SupplierID: 1,
    CategoryID: 1,
    QuantityPerUnit: '10 boxes x 20 bags',
    UnitPrice: 18,
    UnitsInStock: 39,
    UnitsOnOrder: 0,
    ReorderLevel: 10,
    Discontinued: 0
  },
  {
    _id: ObjectId("6006b12f54d343556918c11e"),
    ProductID: 9,
    ProductName: 'Mishi Kobe Niku',
    SupplierID: 4,
    CategoryID: 6,
    QuantityPerUnit: '18 - 500 g pkgs.',
    UnitPrice: 97,
    UnitsInStock: 29,
    UnitsOnOrder: 0,
    ReorderLevel: 0,
    Discontinued: 1
  }
]
```

Limiter le nombre des documents à afficher à 2 ;

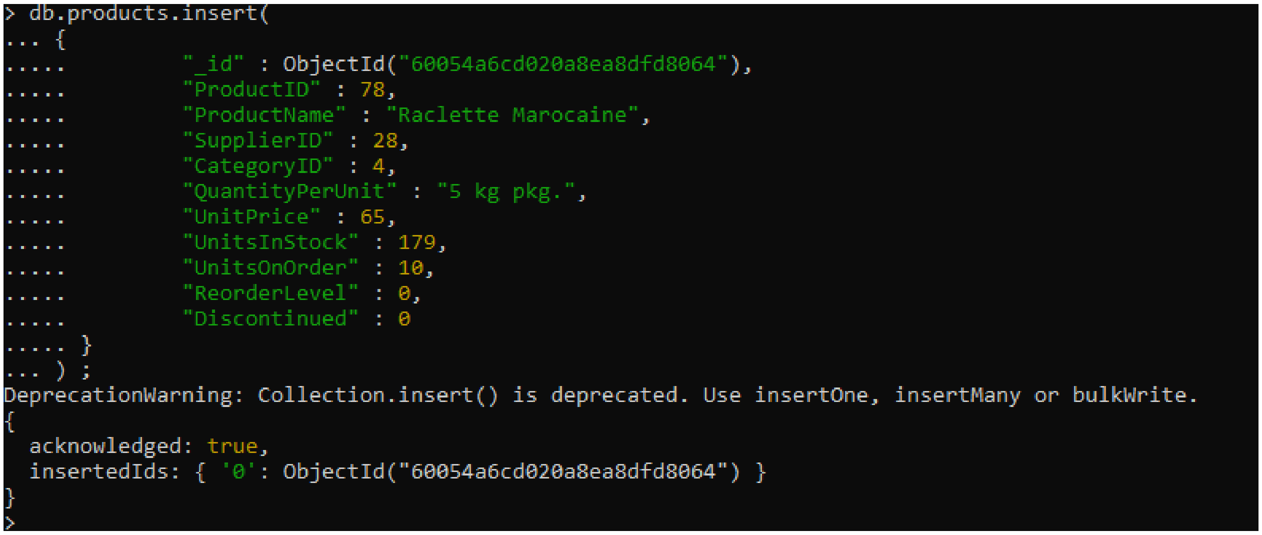
db.products.find().sort({ProductID:-1}).limit(1).pretty();

La commande affiche document limiter par l'ordre descendant de ProductID



```
> db.products.find().sort({ProductID:-1}).limit(1).pretty();
[
  {
    _id: ObjectId("6006b12f54d343556918c168"),
    ProductID: 77,
    ProductName: 'Original Frankfurter grüne Soße',
    SupplierID: 12,
    CategoryID: 2,
    QuantityPerUnit: '12 boxes',
    UnitPrice: 13,
    UnitsInStock: 32,
    UnitsOnOrder: 0,
    ReorderLevel: 15,
    Discontinued: 0
  }
]
```

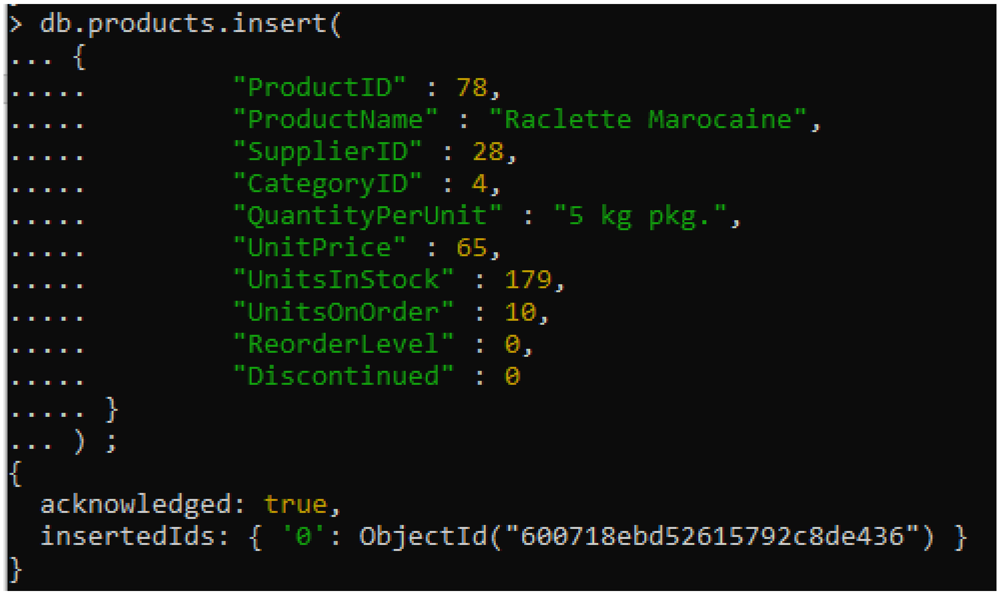
Exécuter la commande



```
> db.products.insert(
... {
.....         "_id" : ObjectId("60054a6cd020a8ea8dfd8064"),
.....         "ProductID" : 78,
.....         "ProductName" : "Raclette Marocaine",
.....         "SupplierID" : 28,
.....         "CategoryID" : 4,
.....         "QuantityPerUnit" : "5 kg pkg.",
.....         "UnitPrice" : 65,
.....         "UnitsInStock" : 179,
.....         "UnitsOnOrder" : 10,
.....         "ReorderLevel" : 0,
.....         "Discontinued" : 0
..... }
... ) ;
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("60054a6cd020a8ea8dfd8064") }
}
>
```

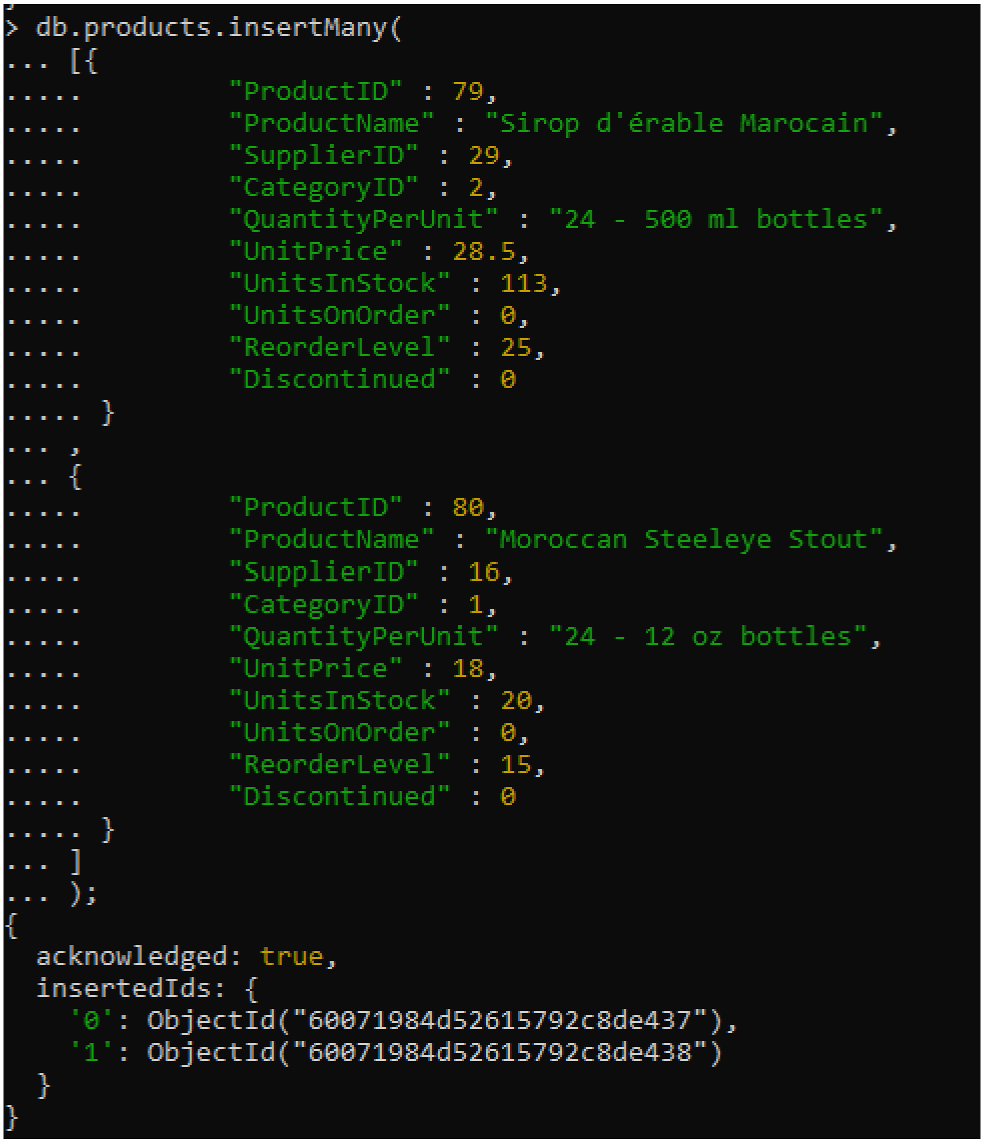
Remarque : on a spécifié Id

Exécuter la commande



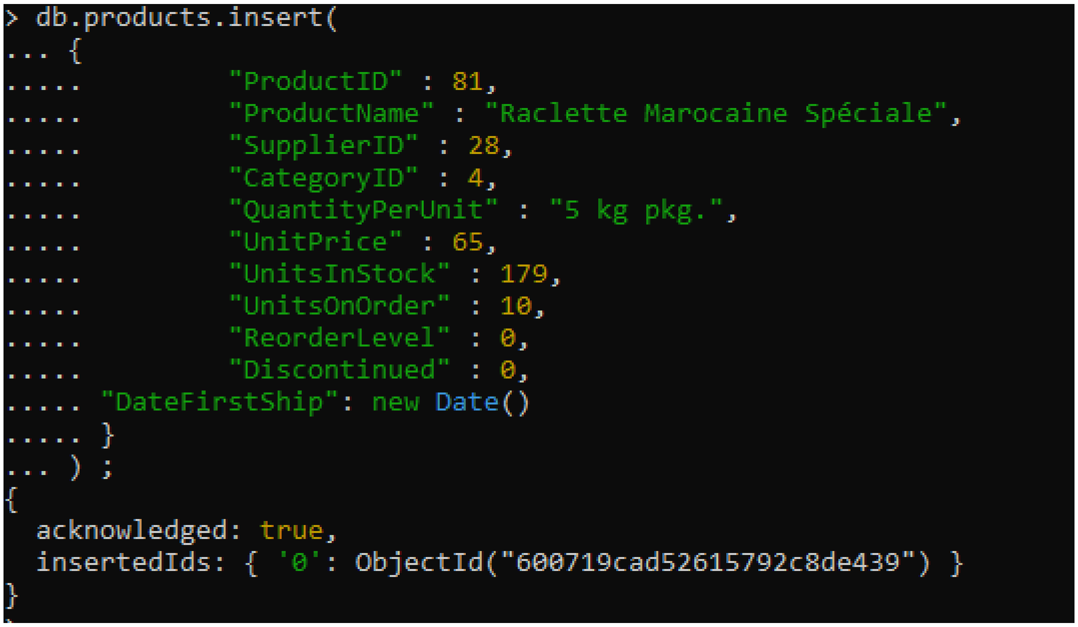
```
> db.products.insert(
... {
.....         "ProductID" : 78,
.....         "ProductName" : "Raclette Marocaine",
.....         "SupplierID" : 28,
.....         "CategoryID" : 4,
.....         "QuantityPerUnit" : "5 kg pkg.",
.....         "UnitPrice" : 65,
.....         "UnitsInStock" : 179,
.....         "UnitsOnOrder" : 10,
.....         "ReorderLevel" : 0,
.....         "Discontinued" : 0
..... }
... ) ;
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("600718ebd52615792c8de436") }
}
```

Exécuter la commande



```
> db.products.insertMany(
... [{
.....         "ProductID" : 79,
.....         "ProductName" : "Sirop d'érable Marocain",
.....         "SupplierID" : 29,
.....         "CategoryID" : 2,
.....         "QuantityPerUnit" : "24 - 500 ml bottles",
.....         "UnitPrice" : 28.5,
.....         "UnitsInStock" : 113,
.....         "UnitsOnOrder" : 0,
.....         "ReorderLevel" : 25,
.....         "Discontinued" : 0
..... }
... ,
... {
.....         "ProductID" : 80,
.....         "ProductName" : "Moroccan Steeleye Stout",
.....         "SupplierID" : 16,
.....         "CategoryID" : 1,
.....         "QuantityPerUnit" : "24 - 12 oz bottles",
.....         "UnitPrice" : 18,
.....         "UnitsInStock" : 20,
.....         "UnitsOnOrder" : 0,
.....         "ReorderLevel" : 15,
.....         "Discontinued" : 0
..... }
... ]
... );
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("60071984d52615792c8de437"),
    '1': ObjectId("60071984d52615792c8de438")
  }
}
```

Exécuter la commande

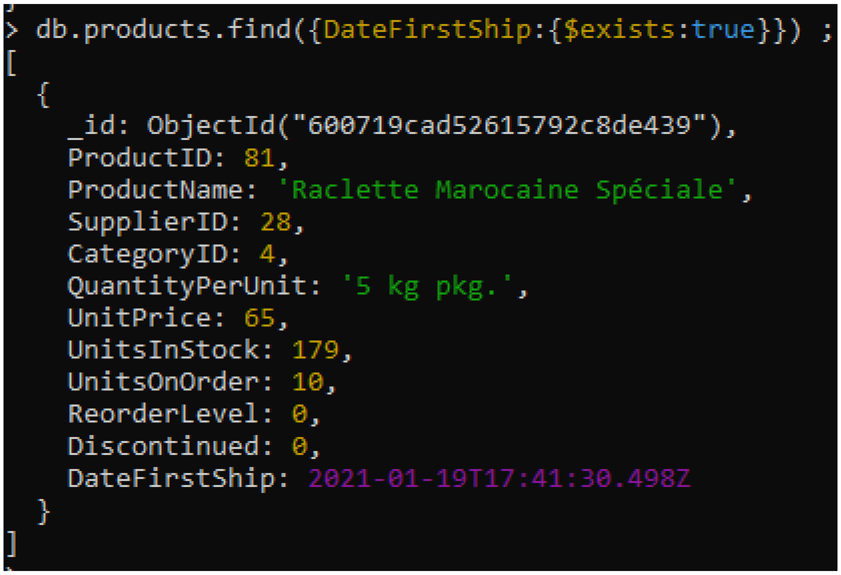


```
> db.products.insert(
... {
.....           "ProductID" : 81,
.....           "ProductName" : "Raclette Marocaine Spéciale",
.....           "SupplierID" : 28,
.....           "CategoryID" : 4,
.....           "QuantityPerUnit" : "5 kg pkg.",
.....           "UnitPrice" : 65,
.....           "UnitsInStock" : 179,
.....           "UnitsOnOrder" : 10,
.....           "ReorderLevel" : 0,
.....           "Discontinued" : 0,
..... "DateFirstShip": new Date()
..... }
... ) ;
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("600719cad52615792c8de439") }
}
```

Déférence entre ce document et les documents déjà inséré est : on a ajouté la fonction DateFirstShip pour ajouter la date de document

Exécuter la commande

db.products.find({DateFirstShip:{$exists:true}})

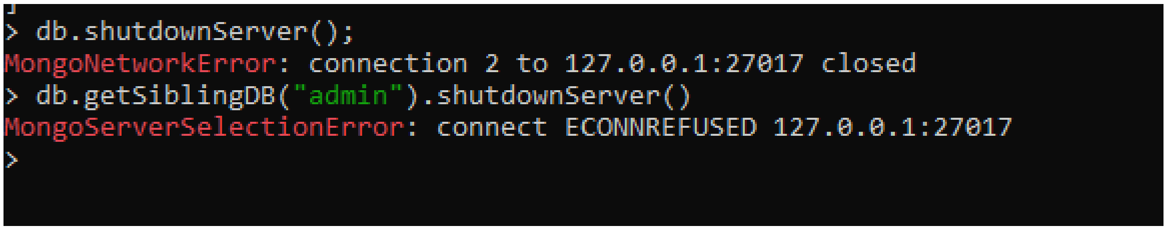


```
> db.products.find({DateFirstShip:{$exists:true}}) ;
[
  {
    _id: ObjectId("600719cad52615792c8de439"),
    ProductID: 81,
    ProductName: 'Raclette Marocaine Spéciale',
    SupplierID: 28,
    CategoryID: 4,
    QuantityPerUnit: '5 kg pkg.',
    UnitPrice: 65,
    UnitsInStock: 179,
    UnitsOnOrder: 10,
    ReorderLevel: 0,
    Discontinued: 0,
    DateFirstShip: 2021-01-19T17:41:30.498Z
  }
]
```

Renvoi la date de création de document

Exécuter la commande

db.shutdownServer();



```
> db.shutdownServer();
MongoNetworkError: connection 2 to 127.0.0.1:27017 closed
> db.getSiblingDB("admin").shutdownServer()
MongoServerSelectionError: connect ECONNREFUSED 127.0.0.1:27017
>
```