

TP 5 MONGODB

djennaoui raouf 2014 0000 2338

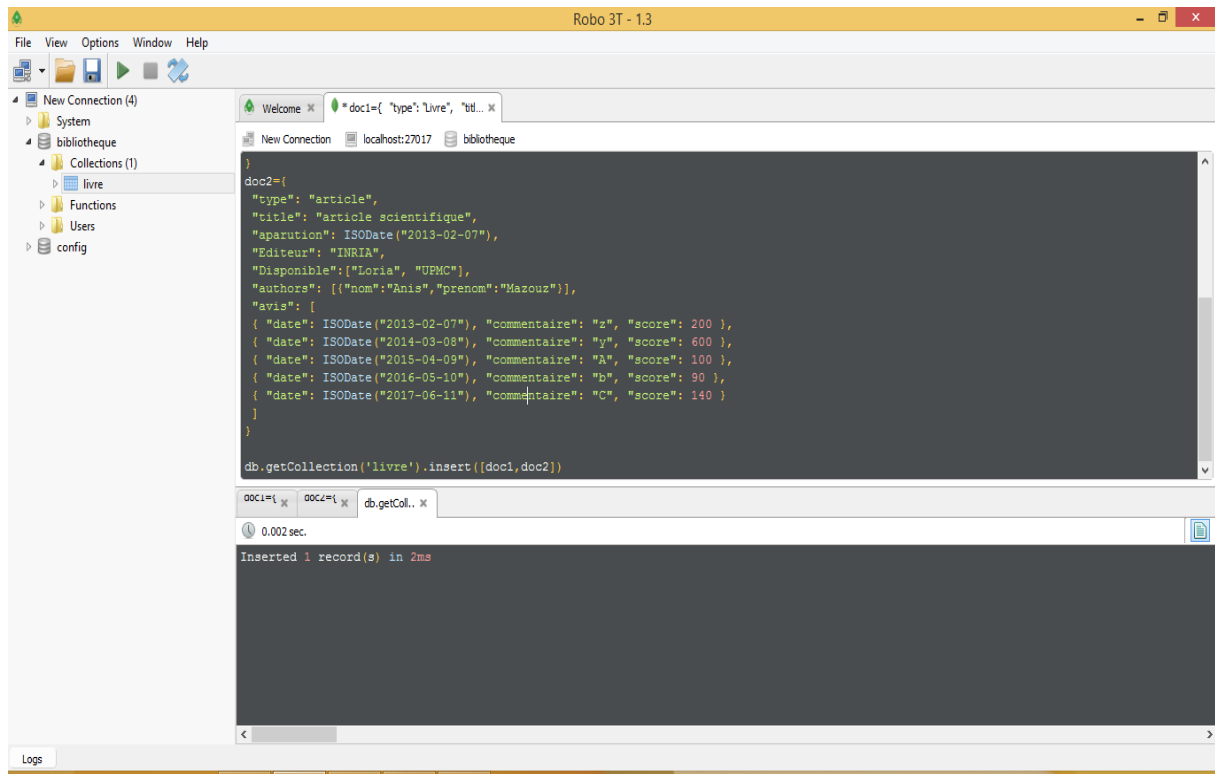
1)

```
doc1={
  "type": "Livre",
  "title": "Modern Database Systems: The Object Model, Interoperability, and Beyond",
  "apparition": ISODate("2012-02-07"),
  "Editeur": "Eyrolles",
  "Disponible":["Amazon", "Fnac"],
  "authors": [{"nom":"Youcef","prenom":"Benkheda"}],
  "avis": [
    { "date": ISODate("2013-02-07"), "commentaire": "A", "score": 2 },
    { "date": ISODate("2014-03-08"), "commentaire": "C", "score": 6 },
    { "date": ISODate("2015-04-09"), "commentaire": "A", "score": 10 },
    { "date": ISODate("2016-05-10"), "commentaire": "b", "score": 9 },
    { "date": ISODate("2017-06-11"), "commentaire": "C", "score": 14 }
  ]
}

doc2={
  "type": "article",
  "title": "article scientifique",
  "aparution": ISODate("2013-02-07"),
  "Editeur": "INRIA",
  "Disponible":["Loria", "UPMC"],
  "authors": [{"nom":"Anis","prenom":"Mazouz"}],
  "avis": [
    { "date": ISODate("2013-02-07"), "commentaire": "z", "score": 200 },
    { "date": ISODate("2014-03-08"), "commentaire": "y", "score": 600 },
    { "date": ISODate("2015-04-09"), "commentaire": "A", "score": 100 },
    { "date": ISODate("2016-05-10"), "commentaire": "b", "score": 90 },
    { "date": ISODate("2017-06-11"), "commentaire": "C", "score": 140 }
  ]
}
```

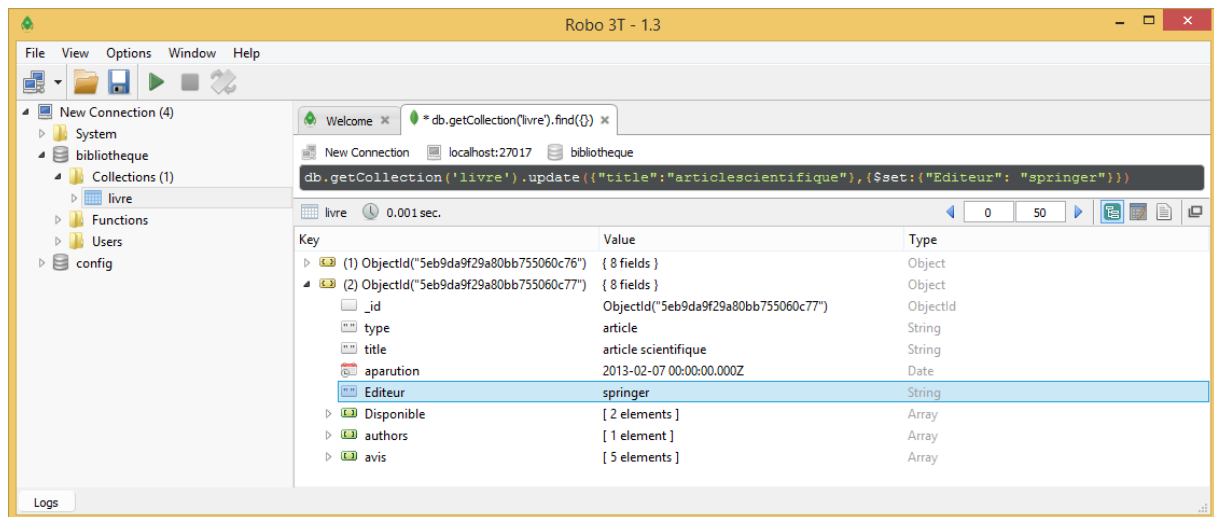
```
db.getCollection('livre').insert([doc1,doc2])
```

insertion des deux variable doc1 et doc2 voir la figure suivante:



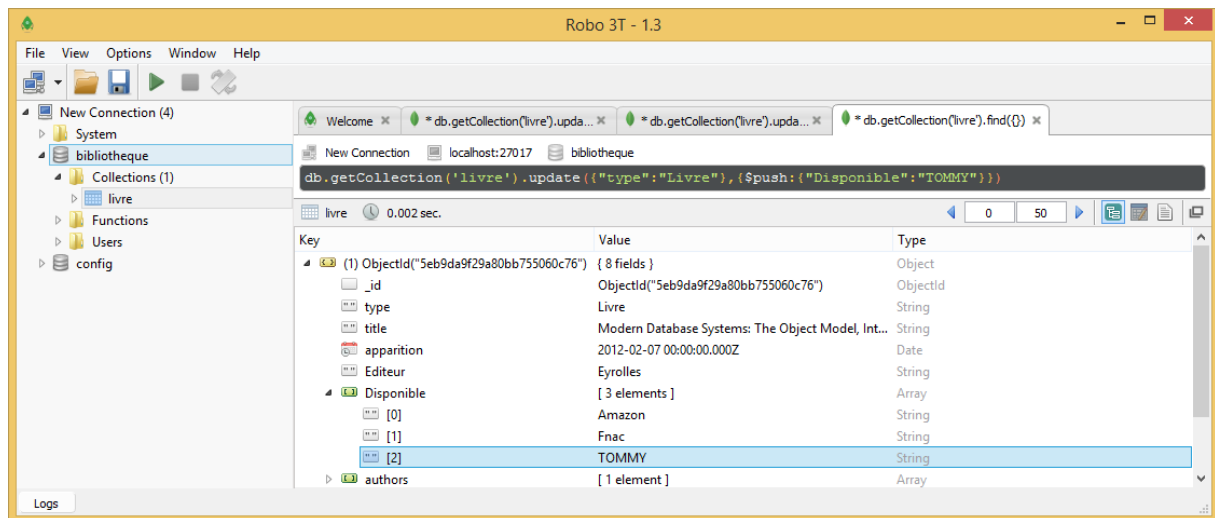
2)

`db.getCollection('livre').update({"title":"article scientifique"},{$set:{"Editeur":"springer"}})`



3)

```
db.getCollection('livre').update({"type":"Livre"},{$push:{"Disponible":"TOMMY"}})
```

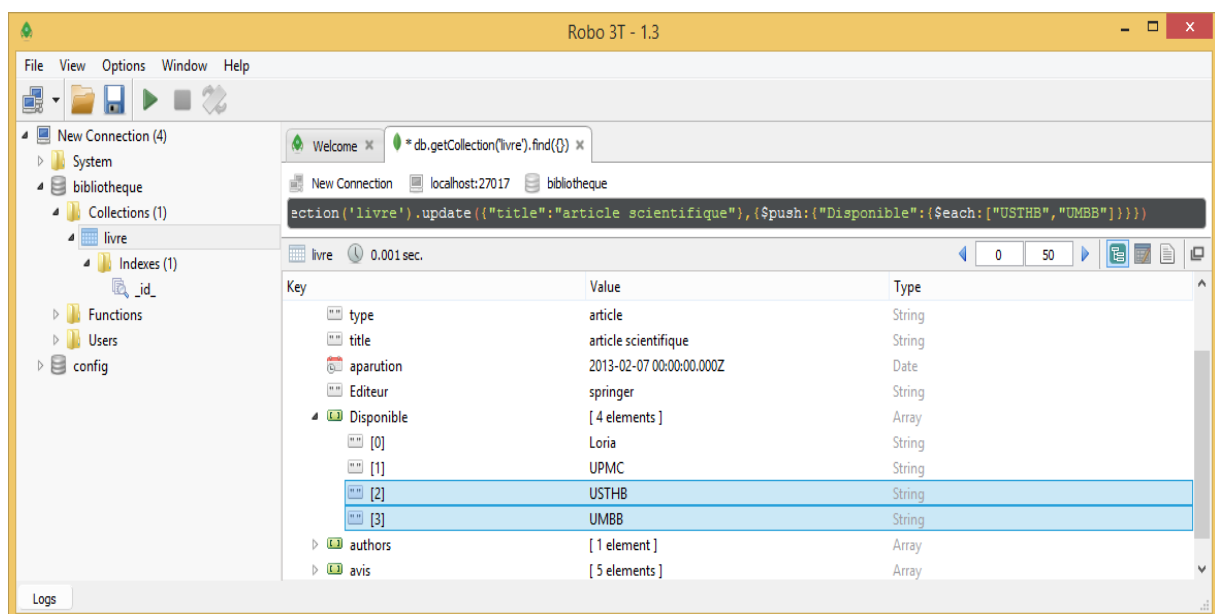


4)

```
db.getCollection('livre').update({"title":"article  
scientifique"},{$addToSet:{"Disponible":{"$each:["USTHB","UMBB"]}}})
```

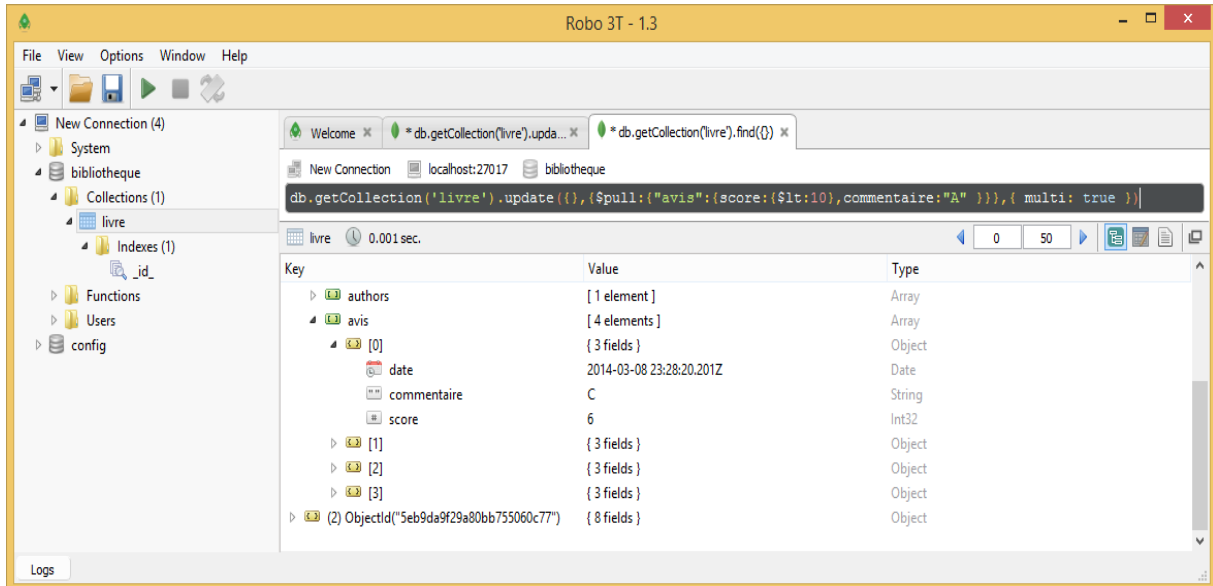
ou bien

```
db.getCollection('livre').update({"title":"article  
scientifique"},{$push:{"Disponible":{"$each:["USTHB","UMBB"]}}})
```



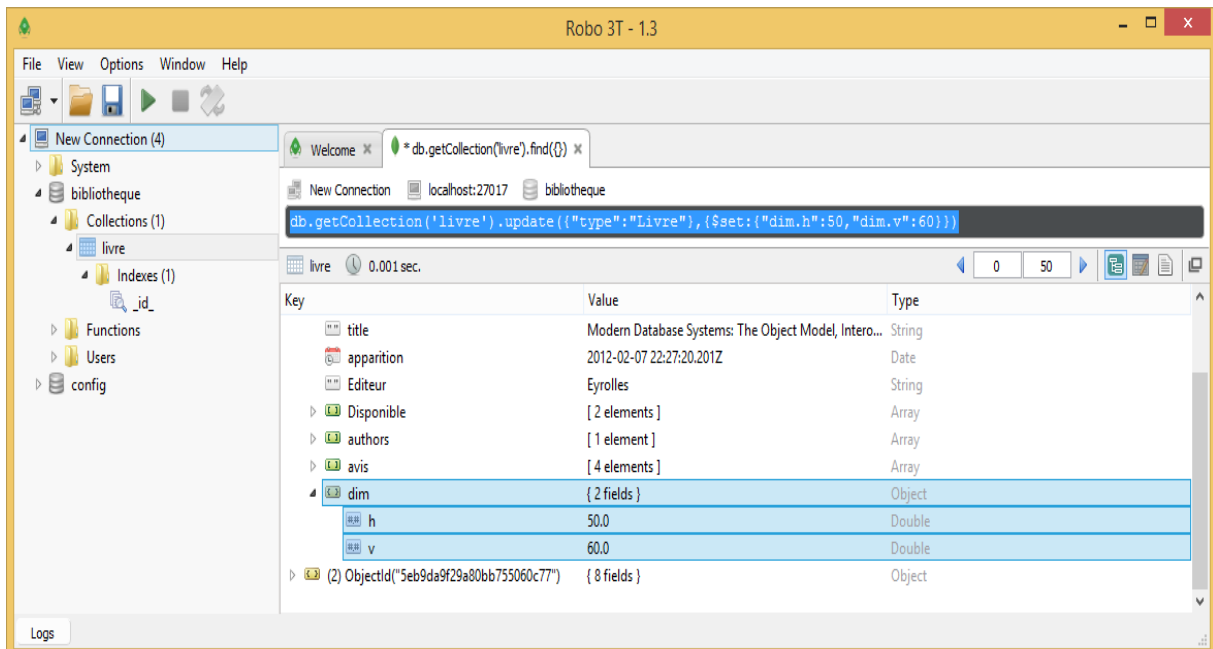
5)

```
db.getCollection('livre').update({},{$pull:{"avis":{"score:{$lt:10},commentaire:"A" }}}},{
multi: true })
```



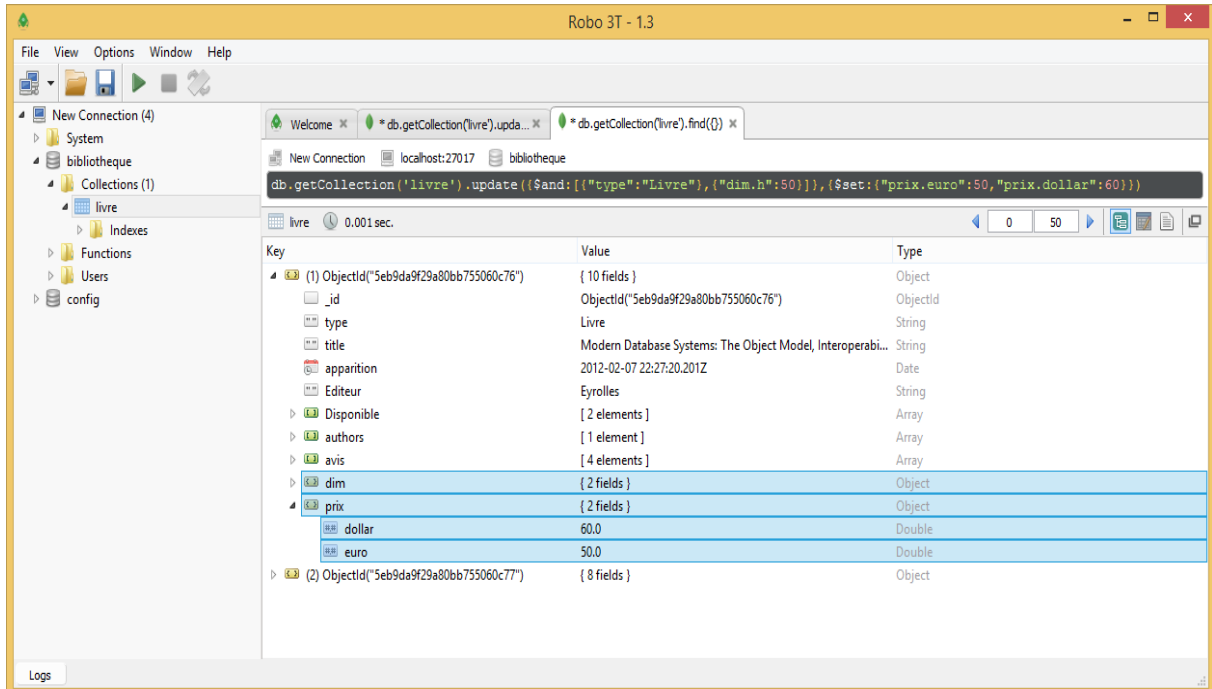
6)

```
db.getCollection('livre').update({"type":"Livre"},{$set:{"dim.h":50,"dim.v":60}})
```



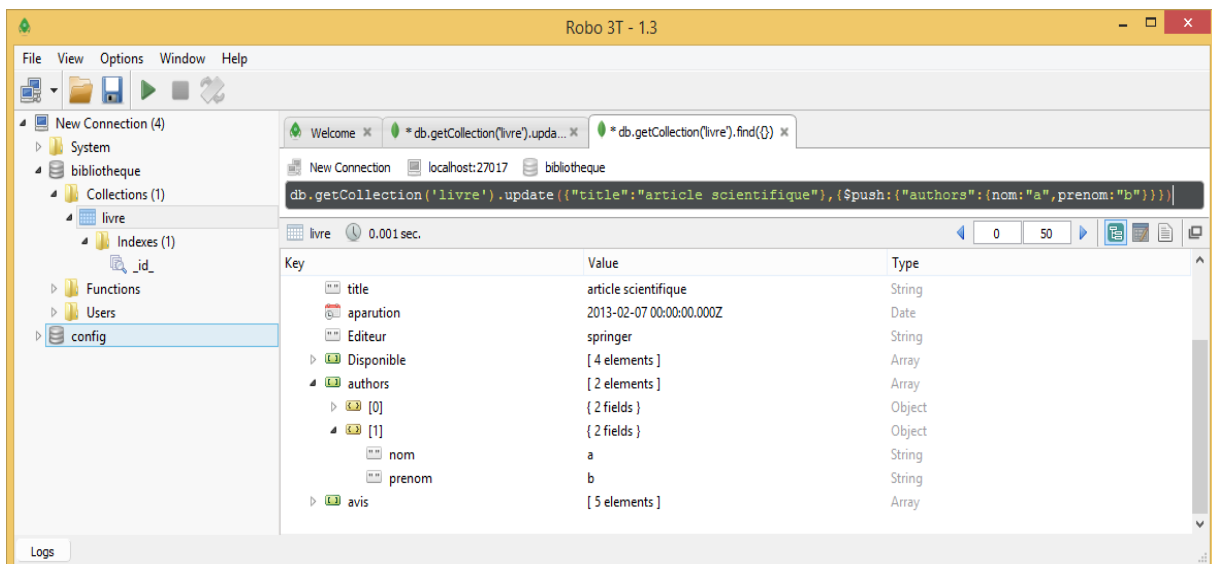
7)

```
db.getCollection('livre').update({'$and':{'type':'Livre'},{'dim.h':50}},{'$set':{'prix.euro':50,'prix.dollar':60}})
```



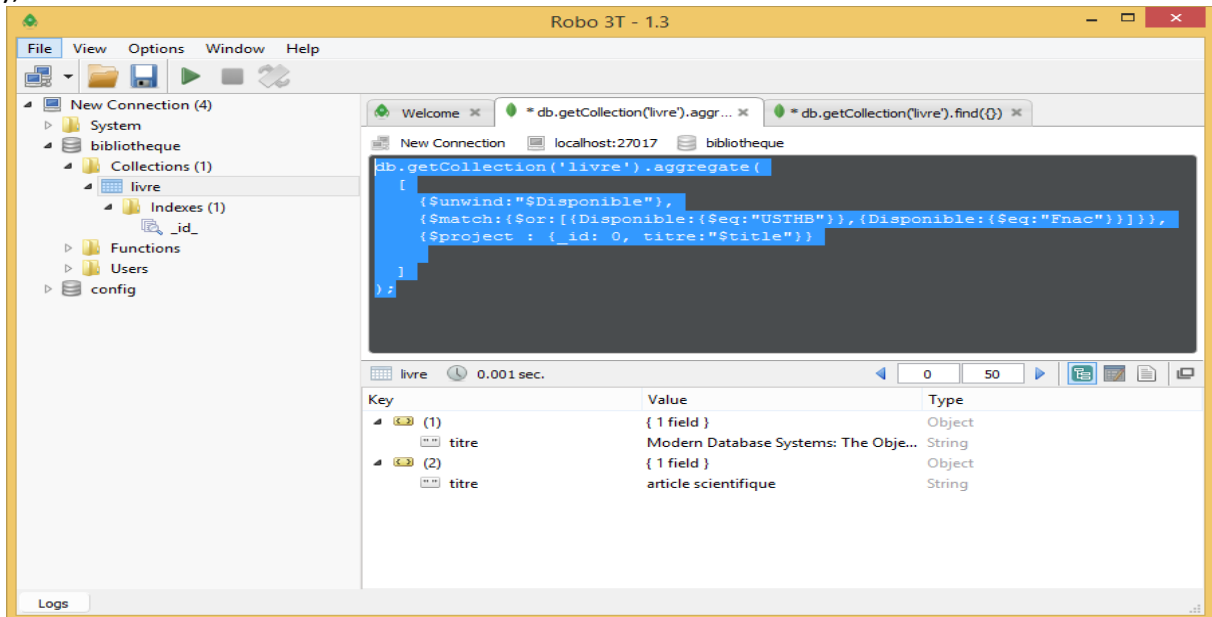
8)

```
db.getCollection('livre').update({'title':'article scientifique'},{'$push':{'authors':{'nom':'a','prenom':'b'}}})
```



9)

```
db.getCollection('livre').aggregate(  
  [  
    {$unwind:"$Disponible"},  
    {$match:{$or: [{Disponible:{$eq:"USTHB"}}, {Disponible:{$eq:"Fnac"}}]}},  
    {$project : {_id: 0, titre:"$titre"}}  
  ]  
);
```



remarque:

\$unwind

Déconstruit un champ de tableau à partir des documents d'entrée pour sortir un document pour *chaque* élément. Chaque document de sortie est le document d'entrée avec la valeur du champ tableau remplacée par l'élément.

10)

```
db.getCollection('livre').aggregate(  
  [  
    {$unwind:"$avis"},  
    {$match:{type:{$eq:"Livre"}}},  
    {$project : {_id: 0, avis:"$avis"}},  
    {$match:{"avis.date":{$gt:ISODate("2015-10-01")}}},  
  ]  
);
```

Robo 3T - 1.3

File View Options Window Help

New Connection (4)

- System
- bibliotheque
 - Collections (1)
 - livre
 - Indexes (1)
 - Functions (0)
 - Users
 - config

Welcome x *db.getCollection('livre').aggre... x

New Connection localhost:27017 bibliotheque

```
db.getCollection('livre').aggregate([
  { $unwind: "$avis" },
  { $match: { type: { $eq: "Livre" } } },
  { $project: { _id: 0, avis: "$avis" } },
  { $match: { "avis.date": { $gt: ISODate("2015-10-01") } } },
])
```

livre 0.002 sec.

Key	Value	Type
(1)	{ 1 field }	Object
avis	{ 3 fields }	Object
date	2016-05-11 01:30:20.201Z	Date
commentaire	b	String
score	9	Int32
(2)	{ 1 field }	Object
avis	{ 3 fields }	Object
date	2017-06-12 02:31:20.201Z	Date
commentaire	C	String
score	14	Int32

Logs

11)

`db.getCollection('livre').update({"type":"Livre"},{$unset:{"dim":""}},{multi:true})`

Robo 3T - 1.3

File View Options Window Help

New Connection (4)

- System
- bibliotheque
 - Collections (1)
 - livre
 - Indexes
 - Functions
 - Users
 - config

Welcome x *db.getCollection('livre').find({}) x

New Connection localhost:27017 bibliotheque

```
db.getCollection('livre').update({"type":"Livre"},{$unset:{"dim":""}},{multi:true})
```

livre 0.001 sec.

Key	Value	Type
(1)	ObjectId("5eb9da9f29a80bb75... { 9 fields }	Object
_id	ObjectId("5eb9da9f29a80bb755060c76")	ObjectId
type	Livre	String
title	Modern Database Systems: The Object M...	String
apparition	2012-02-07 22:27:20.201Z	Date
Editeur	Eyrolles	String
Disponible	[2 elements]	Array
authors	[1 element]	Array
avis	[4 elements]	Array
prix	{ 2 fields }	Object
(2)	ObjectId("5eb9da9f29a80bb75... { 8 fields }	Object

Logs

13)

```
db.getCollection('livre').aggregate([
  { $group : { _id : "$type", titre:{$push:"$titre" } } },
  { $out:"doc" }
])
```

The screenshot shows the Robo 3T - 1.3 application window. The left sidebar displays a tree view of the database structure, including 'System', 'bibliotheque', 'Collections (2)' (with 'doc' and 'livre'), 'Functions', 'Users', and 'config'. The main window has a tab for the query: '* db.getCollection('doc').find({}) *'. The query editor contains the following aggregation query:

```
db.getCollection('livre').aggregate([
  { $group : { _id : "$type", titre:{$push:"$titre" } } },
  { $out:"doc" }
])
```

Below the query editor, the results are displayed in a table format. The table has three columns: 'Key', 'Value', and 'Type'. The results are as follows:

Key	Value	Type
(1) article	{ 2 fields }	Object
_id	article	String
titre	[1 element]	Array
[0]	article scientifique	String
(2) Livre	{ 2 fields }	Object
_id	Livre	String
titre	[2 elements]	Array
[0]	Modern Database Systems: The Object M...	String
[1]	new livre	String

The bottom of the window shows a 'Logs' tab.