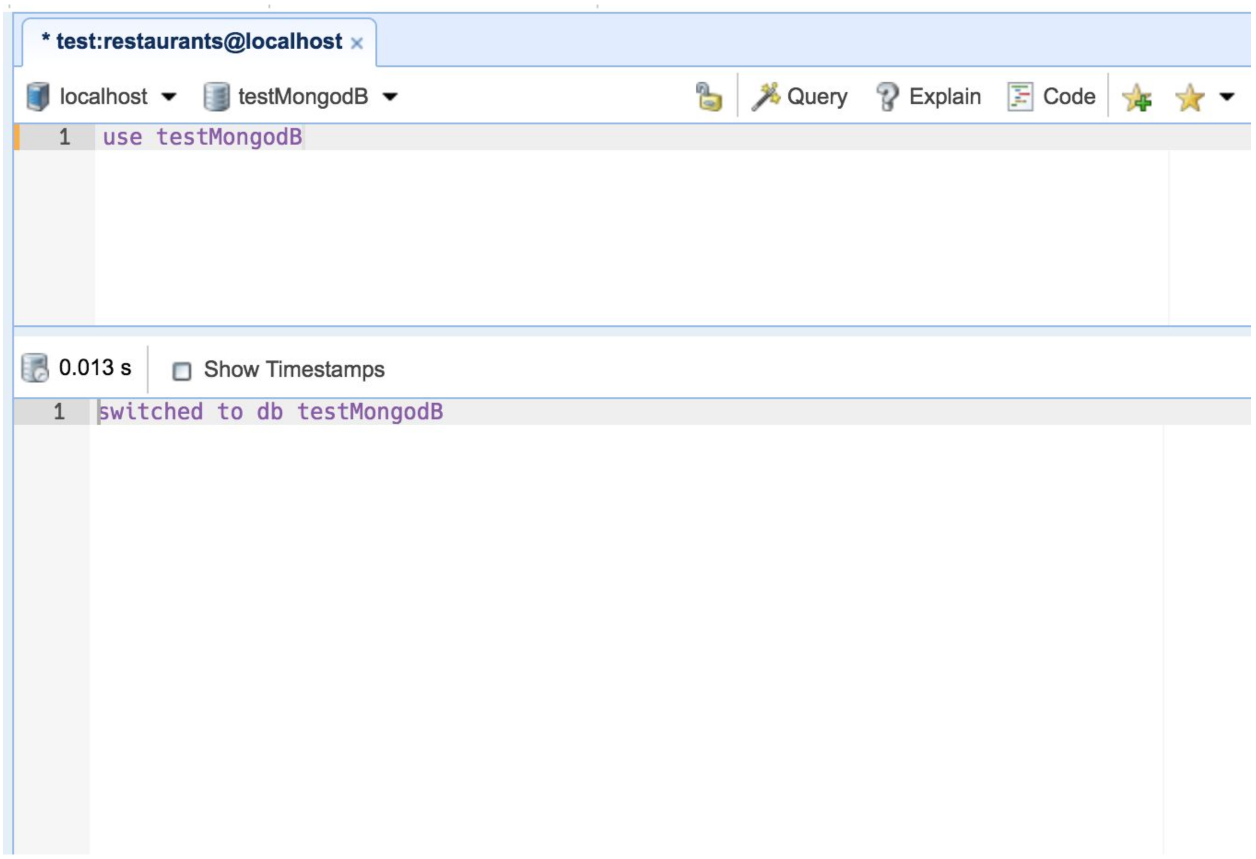
Task 1

Operation 1:



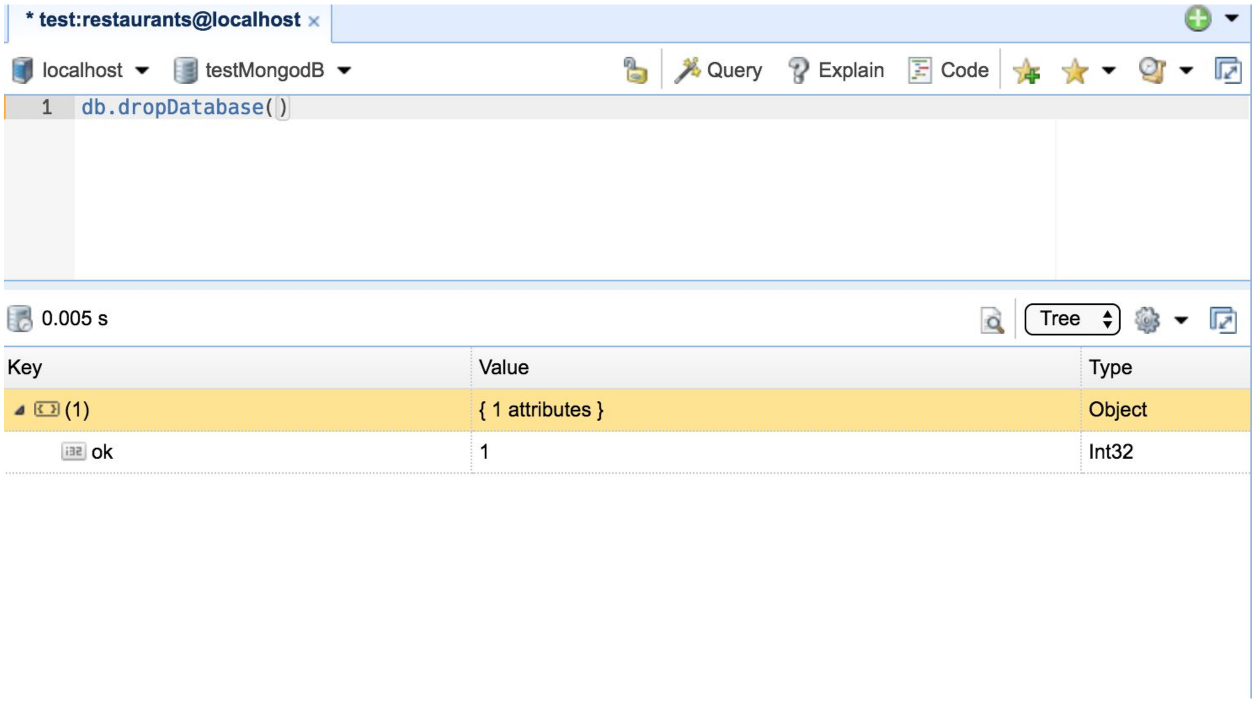
```
use testMongodB
```

0.013 s

Show Timestamps

```
switched to db testMongodB
```

Operation 2:

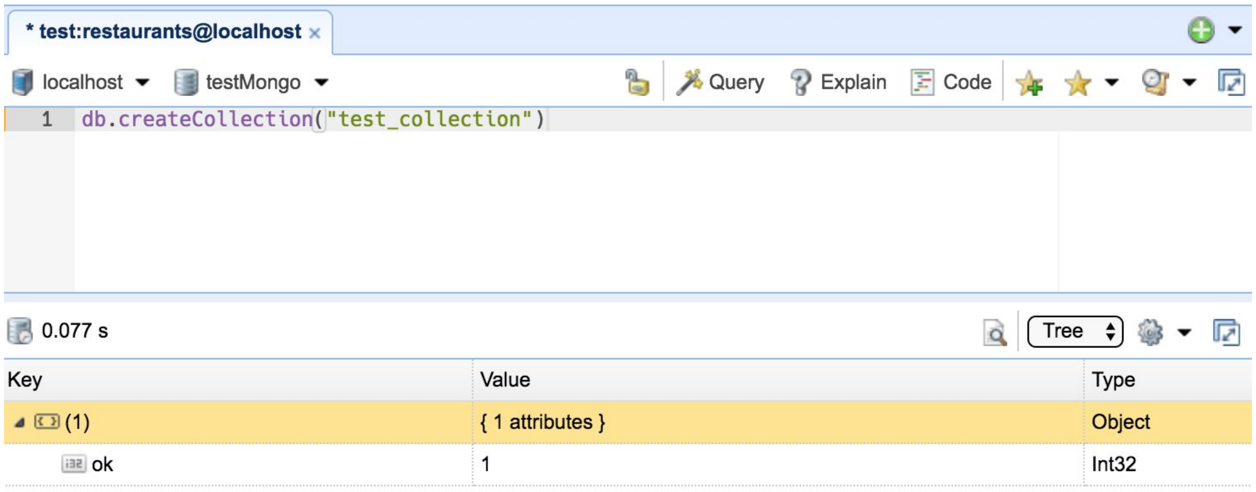


```
db.dropDatabase()
```

0.005 s

Key	Value	Type
(1)	{ 1 attributes }	Object
ok	1	Int32

Operation 3:

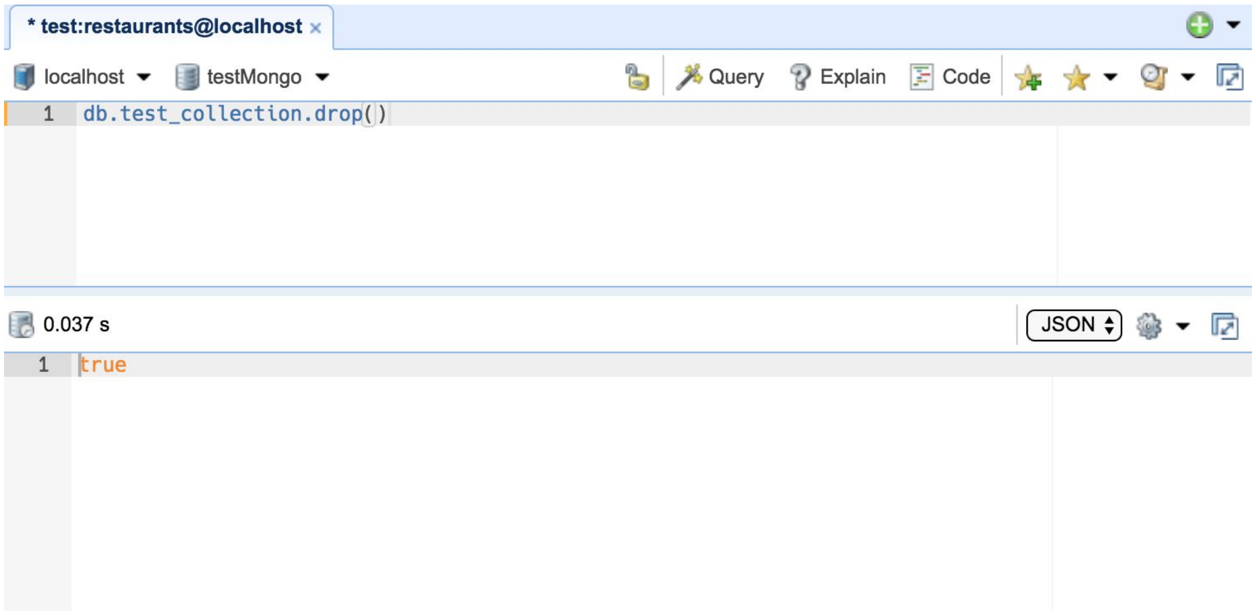


```
db.createCollection("test_collection")
```

0.077 s

Key	Value	Type
(1)	{ 1 attributes }	Object
ok	1	Int32

Operation 4:

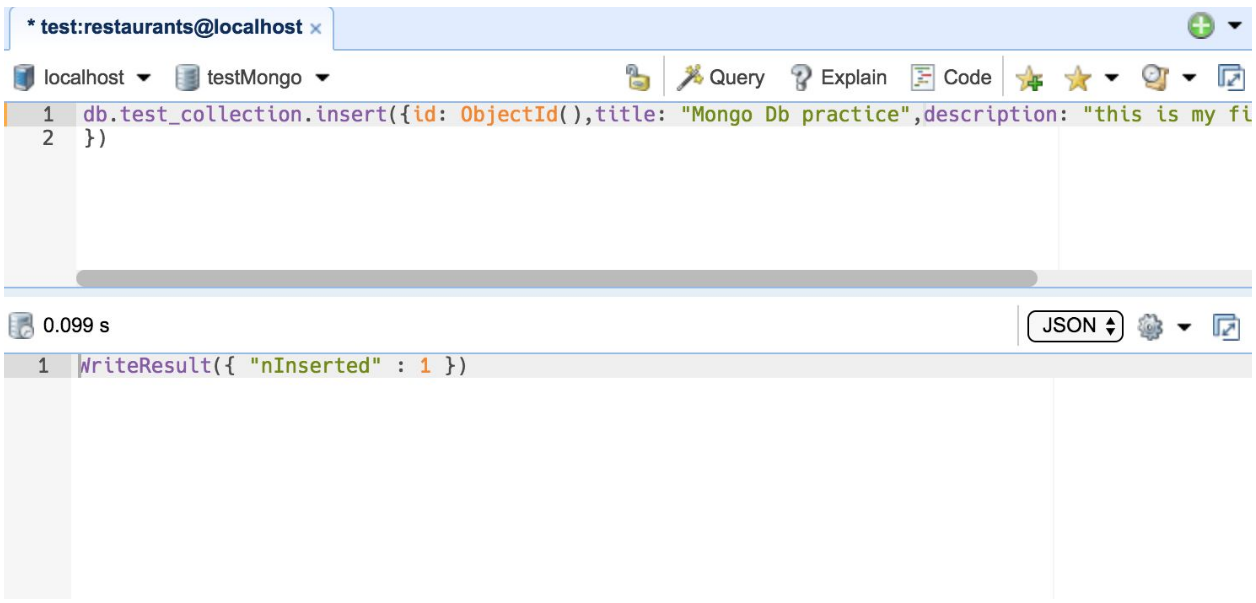


```
db.test_collection.drop()
```

0.037 s

```
true
```

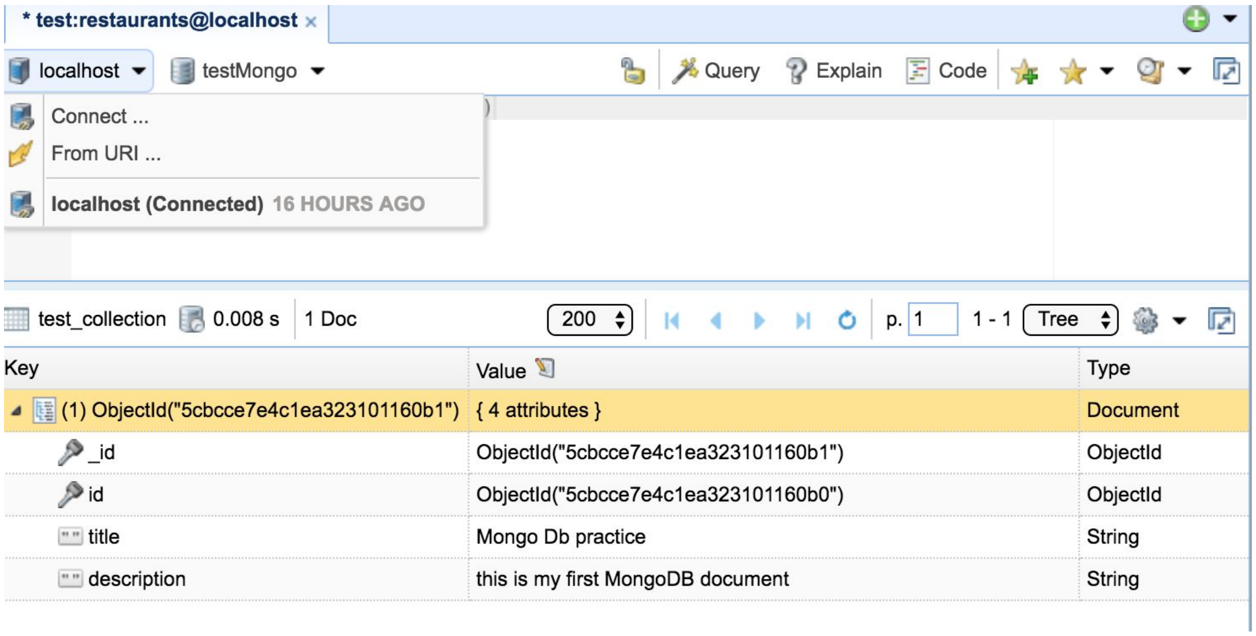
Operation 5:



```
db.test_collection.insert({id: ObjectId(),title: "Mongo Db practice",description: "this is my fi
})
```

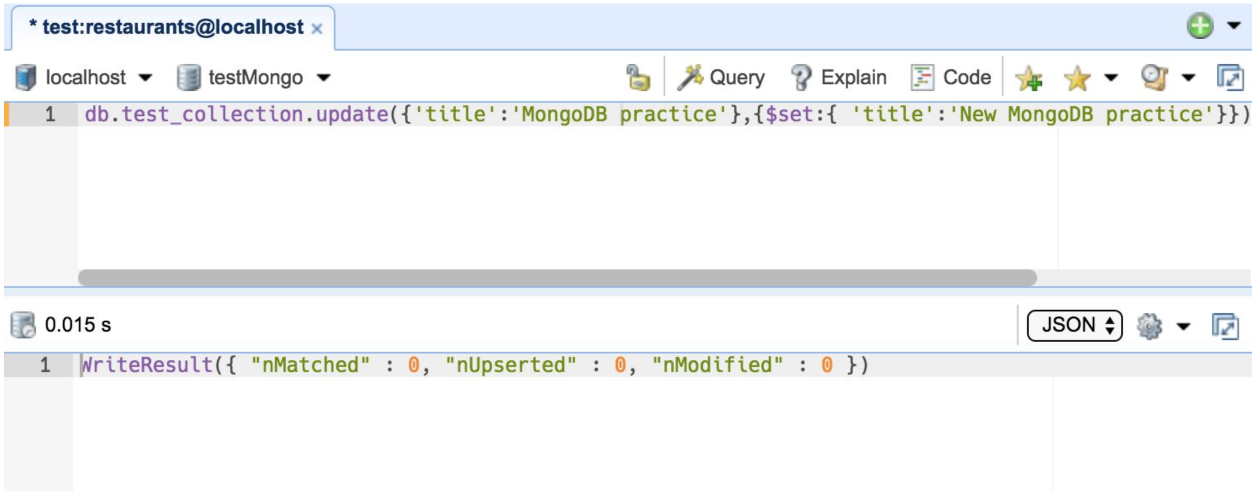
```
WriteResult({ "nInserted" : 1 })
```

Operation 6:



Key	Value	Type
(1) ObjectId("5cbcce7e4c1ea323101160b1")	{ 4 attributes }	Document
_id	ObjectId("5cbcce7e4c1ea323101160b1")	ObjectId
id	ObjectId("5cbcce7e4c1ea323101160b0")	ObjectId
title	Mongo Db practice	String
description	this is my first MongoDB document	String

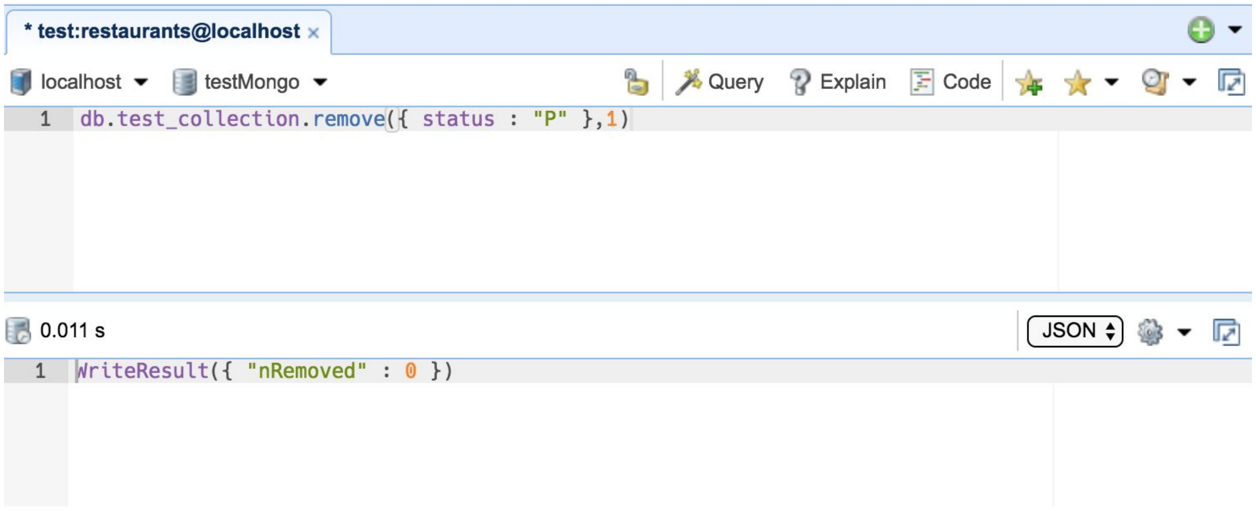
Operation 7:



```
db.test_collection.update({'title':'MongoDB practice'},{$set:{ 'title':'New MongoDB practice'}})
```

```
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
```

Operation 8:



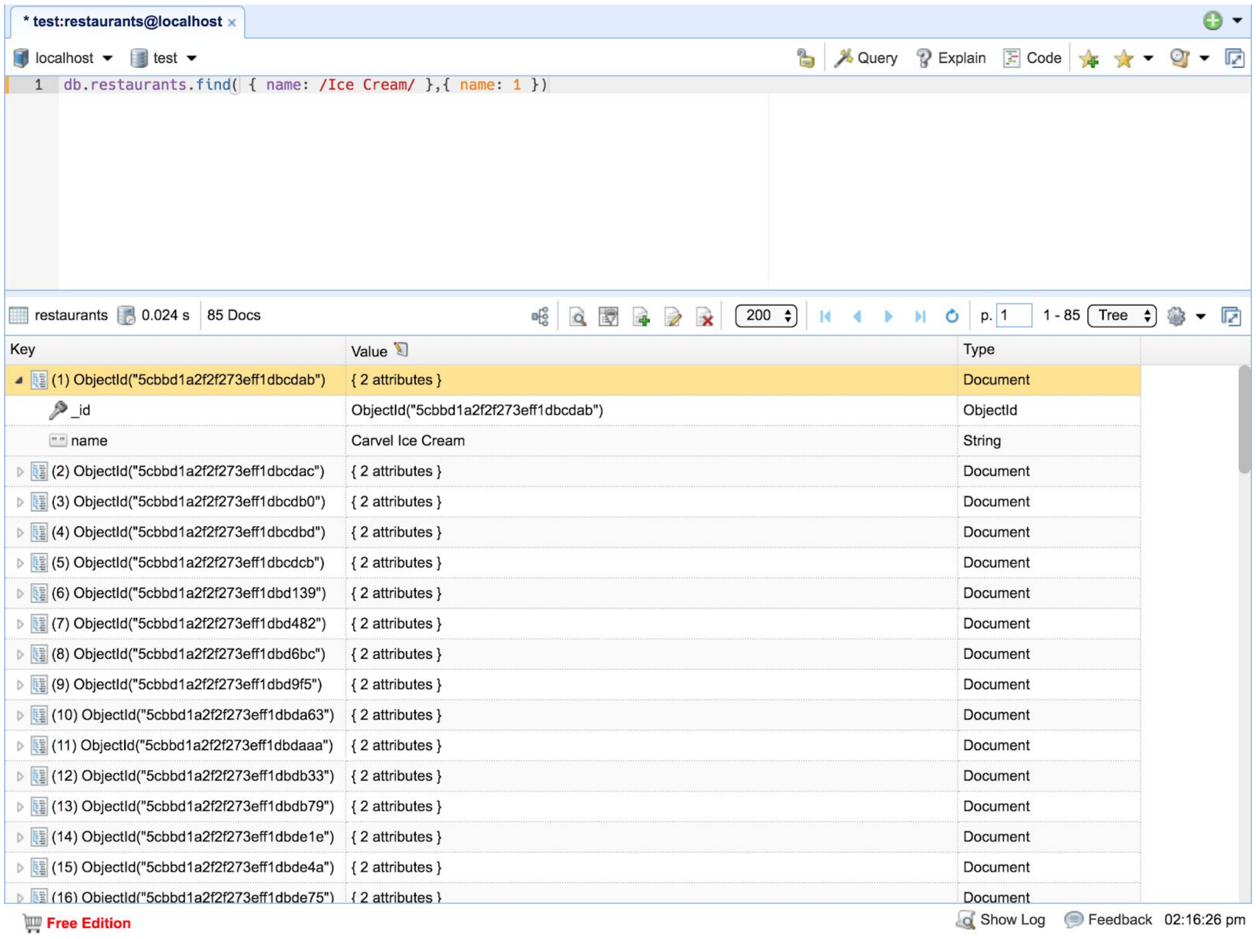
```
db.test_collection.remove({ status : "P" },1)
```

```
WriteResult({ "nRemoved" : 0 })
```

Task 2

Query 1:

db.restaurants.find({ name: /Ice Cream/ },{ name: 1 })



* test:restaurants@localhost

localhost test | Query | Explain | Code

```
db.restaurants.find( { name: /Ice Cream/ },{ name: 1 })
```

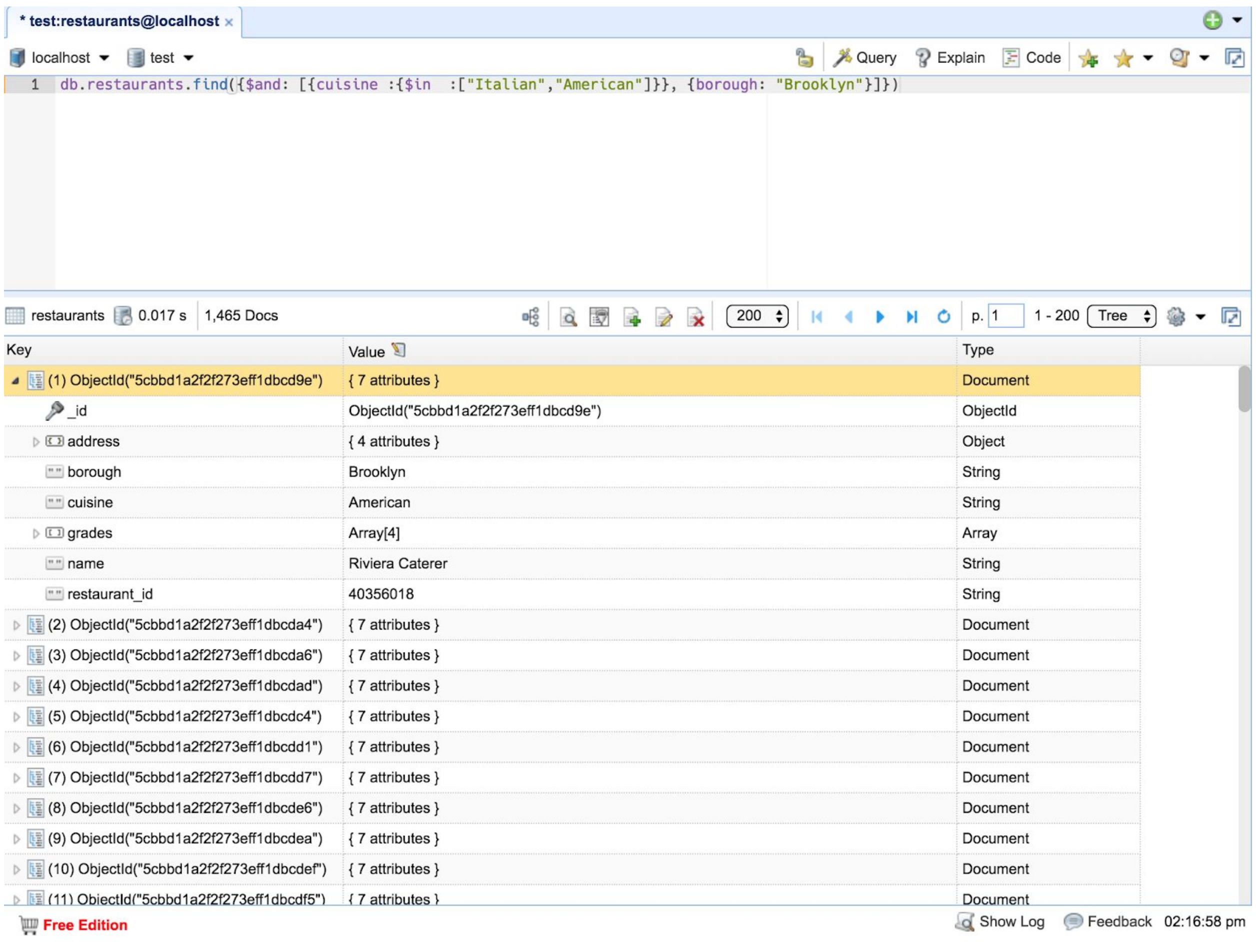
restaurants 0.024 s 85 Docs | 200 | p. 1 | 1 - 85 | Tree

Key	Value	Type
(1) ObjectId("5cbbd1a2f2f273eff1dbcdab")	{ 2 attributes }	Document
_id	ObjectId("5cbbd1a2f2f273eff1dbcdab")	ObjectId
name	Carvel Ice Cream	String
(2) ObjectId("5cbbd1a2f2f273eff1dbcdac")	{ 2 attributes }	Document
(3) ObjectId("5cbbd1a2f2f273eff1dbcdb0")	{ 2 attributes }	Document
(4) ObjectId("5cbbd1a2f2f273eff1dbcdbd")	{ 2 attributes }	Document
(5) ObjectId("5cbbd1a2f2f273eff1dbcdcb")	{ 2 attributes }	Document
(6) ObjectId("5cbbd1a2f2f273eff1dbd139")	{ 2 attributes }	Document
(7) ObjectId("5cbbd1a2f2f273eff1dbd482")	{ 2 attributes }	Document
(8) ObjectId("5cbbd1a2f2f273eff1dbd6bc")	{ 2 attributes }	Document
(9) ObjectId("5cbbd1a2f2f273eff1dbd9f5")	{ 2 attributes }	Document
(10) ObjectId("5cbbd1a2f2f273eff1dbda63")	{ 2 attributes }	Document
(11) ObjectId("5cbbd1a2f2f273eff1dbdaaa")	{ 2 attributes }	Document
(12) ObjectId("5cbbd1a2f2f273eff1dbdb33")	{ 2 attributes }	Document
(13) ObjectId("5cbbd1a2f2f273eff1dbdb79")	{ 2 attributes }	Document
(14) ObjectId("5cbbd1a2f2f273eff1dbde1e")	{ 2 attributes }	Document
(15) ObjectId("5cbbd1a2f2f273eff1dbde4a")	{ 2 attributes }	Document
(16) ObjectId("5cbbd1a2f2f273eff1dbde75")	{ 2 attributes }	Document

Free Edition | Show Log | Feedback | 02:16:26 pm

Query 2:

```
db.restaurants.find(
   {$and: [{cuisine :{$in  :["Italian","American"]}}, {borough: "Brooklyn"}
      ]
   }
)
```



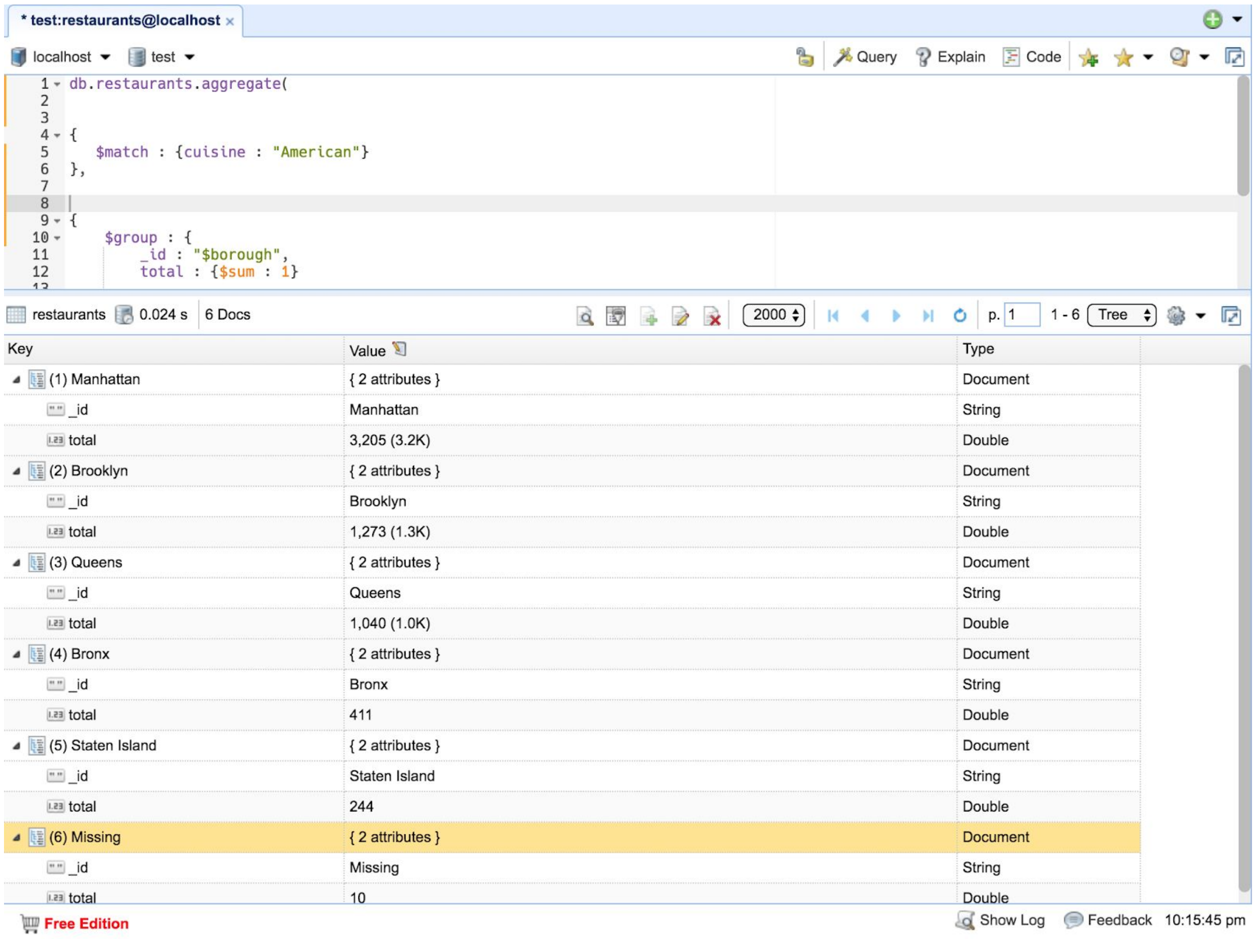
```
db.restaurants.find({$and: [{cuisine :{$in  :["Italian","American"]}}, {borough: "Brooklyn"}]})
```

restaurants 0.017 s 1,465 Docs

```
Key                                          Value                                    Type
(1) ObjectId("5cbbd1a2f2f273eff1dbcd9e")     { 7 attributes }                         Document
    _id                                      ObjectId("5cbbd1a2f2f273eff1dbcd9e")     ObjectId
    address                                  { 4 attributes }                         Object
    borough                                  Brooklyn                                 String
    cuisine                                  American                                 String
    grades                                   Array[4]                                 Array
    name                                     Riviera Caterer                          String
    restaurant_id                            40356018                                 String
(2) ObjectId("5cbbd1a2f2f273eff1dbcda4")     { 7 attributes }                         Document
(3) ObjectId("5cbbd1a2f2f273eff1dbcda6")     { 7 attributes }                         Document
(4) ObjectId("5cbbd1a2f2f273eff1dbcdad")     { 7 attributes }                         Document
(5) ObjectId("5cbbd1a2f2f273eff1dbcdc4")     { 7 attributes }                         Document
(6) ObjectId("5cbbd1a2f2f273eff1dbcdd1")     { 7 attributes }                         Document
(7) ObjectId("5cbbd1a2f2f273eff1dbcdd7")     { 7 attributes }                         Document
(8) ObjectId("5cbbd1a2f2f273eff1dbcde6")     { 7 attributes }                         Document
(9) ObjectId("5cbbd1a2f2f273eff1dbcdea")     { 7 attributes }                         Document
(10) ObjectId("5cbbd1a2f2f273eff1dbcdef")    { 7 attributes }                         Document
(11) ObjectId("5cbbd1a2f2f273eff1dbcdf5")    { 7 attributes }                         Document
```

Query 3:

```
db.restaurants.aggregate(


{
  $match : {cuisine : "American"}
},


{
   $group : {
       _id : "$borough",
       total : {$sum : 1}

   }
}
{
   $sort : { total : -1}
}


)
```

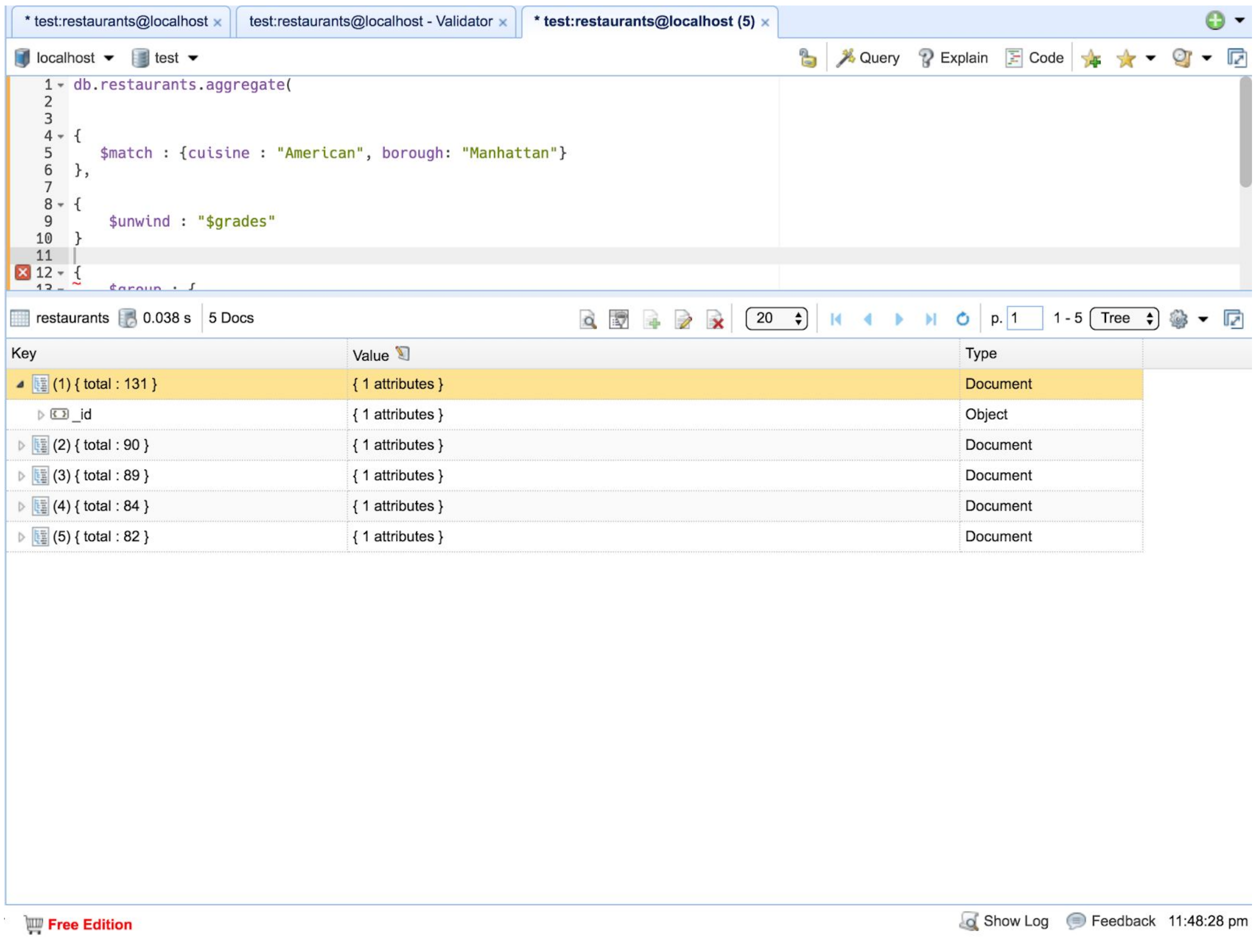


```
db.restaurants.aggregate(


{
   $match : {cuisine : "American"}
},


{
    $group : {
        _id : "$borough",
        total : {$sum : 1}
```

Key	Value	Type
(1) Manhattan	{ 2 attributes }	Document
_id	Manhattan	String
total	3,205 (3.2K)	Double
(2) Brooklyn	{ 2 attributes }	Document
_id	Brooklyn	String
total	1,273 (1.3K)	Double
(3) Queens	{ 2 attributes }	Document
_id	Queens	String
total	1,040 (1.0K)	Double
(4) Bronx	{ 2 attributes }	Document
_id	Bronx	String
total	411	Double
(5) Staten Island	{ 2 attributes }	Document
_id	Staten Island	String
total	244	Double
(6) Missing	{ 2 attributes }	Document
_id	Missing	String
total	10	Double

Query 4:

```
db.restaurants.aggregate(


{
  $match : {cuisine : "American", borough: "Manhattan"}
},


{
   $unwind : "$grades"
}

{
   $group : {
      _id : {
      total : {$sum : "$grades.score"}
      }
}

{
   $sort : { _id : -1}
}


).limit(5)
```



```
db.restaurants.aggregate(


{
   $match : {cuisine : "American", borough: "Manhattan"}
},

{
    $unwind : "$grades"
}

{
    $group : {
```

Key	Value	Type
(1) { total : 131 }	{ 1 attributes }	Document
_id	{ 1 attributes }	Object
(2) { total : 90 }	{ 1 attributes }	Document
(3) { total : 89 }	{ 1 attributes }	Document
(4) { total : 84 }	{ 1 attributes }	Document
(5) { total : 82 }	{ 1 attributes }	Document

Query 5: