Large-Scale and Multi-Structured Databases

Mongo DB Query 1

Prof Pietro Ducange

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ DI PISA

CROSSLAB Innovation for industry 4.0

Copyright Issues

Most of the information included this presentation have been extracted from the official documentation of MongoDB.

Cursor Methods

Loading Data

Load the following collection:

```
db.inventory.insertMany([
   { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
   { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },
   { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
   { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },
   { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }
]);
```

The Find Method and its Cursor

The find() method returns a ***cursor*** to the results.

In the mongo shell, if the returned cursor is not assigned to a variable using the var keyword, the cursor is automatically iterated to access ***up to the first 20 documents*** that match the query.

To manually iterate over the results, assign the returned cursor to a variable with the var keyword, as shown in the following slide.

The mongo shell and the ***drivers*** provide several cursor methods that call on the cursor returned by the find() method to modify its behavior.

Cursor Use: A Simple Example

```
> db.inventory.find();
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160aa"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
> var myCursor = db.inventory.find( );
>
> var myDocument = myCursor.hasNext() ? myCursor.next() : null;
>
> if (myDocument) {
...     var myName = myDocument.item;
...     print (tojson(myName));
... }
"journal"
```

If we update myDocument…

```
> myDocument = myCursor.hasNext() ? myCursor.next() : null;
{
        "_id" : ObjectId("5db99d408215ec9bc20c6c8a"),
        "item" : "notebook",
        "qty" : 50,
        "size" : {
                "h" : 8.5,
                "w" : 11,
                "uom" : "in"
        },
        "status" : "A"
}
> if (myDocument) {     var myName = myDocument.item;     print (tojson(myName)); }
"notebook"
```

Find and Limit

The ***limit()*** method limits the number of documents in the result set. The following operation returns at most 3 documents in the collection inventory.

```
> db.inventory.find();
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160aa"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
> db.inventory.find().limit(3);
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
>
```

Find and Skip

The ***skip()*** method controls the ***starting point*** of the results set. The following operation skips the first 3 documents in the inventory collection and returns all remaining documents:

```
> db.inventory.find();
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160aa"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
> db.inventory.find().limit(3);
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
> db.inventory.find().skip(3);
{ "_id" : ObjectId("5dade2b94a004f52120160aa"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
>
```

Find and Sort

The ***sort()*** method orders the documents in the result set. The following operation returns documents in the inventory collection sorted ***in ascending order*** by the ***qty field***:

```
> db.inventory.find();
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160aa"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
> db.inventory.find().sort({qty:1});
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160aa"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
```

Find and Sort

The following operation returns documents in the inventory collection sorted in ***descending*** order by the ***qty field***:

```
> db.inventory.find().sort({qty:-1});
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160aa"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
```

The following operation returns documents in the inventory collection sorted in ***ascending*** order by the ***status field*** (at first) and in ***descending*** order by the ***qty field:***

```
> db.inventory.find().sort({status:1,qty:-1});
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160aa"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
>
```

Combining Cursor Method

The previous cursor methods can be combined as follows:

```
> db.inventory.find();
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160aa"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
> db.inventory.find().sort({qty:-1});
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160aa"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
> db.inventory.find().sort({qty:-1}).skip(2);
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
>
```

The db.inventory.find().sort({qty:-1}).skip(2) query sorts the collections in ***descending*** order considering the ***qty field*** but ***skips*** the ***first two documents***.

Counting Documents

The method ***db.collection.countDocuments(query, options)*** returns the count of documents that would match a find() query for the collection.

The method ***db.collection.countDocuments does not perform the find()*** operation but instead counts and returns the number of results that match a query.

In Mongo 5 the ***db.collection.count*** has been deprecated**.**

```
exer> db.inventory.find()
[
  {
    _id: ObjectId("6377b1597ceaa1fffab54e6b"),
    item: 'journal',
    qty: 25,
    size: { h: 14, w: 21, uom: 'cm' },
    status: 'A'
  },
  {
    _id: ObjectId("6377b1597ceaa1fffab54e6c"),
    item: 'notebook',
    qty: 50,
    size: { h: 8.5, w: 11, uom: 'in' },
    status: 'A'
  },
  {
    _id: ObjectId("6377b1597ceaa1fffab54e6d"),
    item: 'paper',
    qty: 100,
    size: { h: 8.5, w: 11, uom: 'in' },
    status: 'D'
  },
  {
    _id: ObjectId("6377b1597ceaa1fffab54e6e"),
    item: 'planner',
    qty: 75,
    size: { h: 22.85, w: 30, uom: 'cm' },
    status: 'D'
  },
  {
    _id: ObjectId("6377b1597ceaa1fffab54e6f"),
    item: 'postcard',
    qty: 45,
    size: { h: 10, w: 15.25, uom: 'cm' },
    status: 'A'
  }
]
exer> - db.inventory.countDocuments({status:"A"})
-3
exer>
```

Query on Embedded/Nested Documents

Match an Embedded Document

We can use use the ***query filter*** document { <field>: <value> } where <value> is the document to match.

For example, the following query ***selects all documents*** where the field size equals the document { h: 14, w: 21, uom: "cm" }:

```
> db.inventory.find()
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160aa"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
> db.inventory.find( { size: { h: 14, w: 21, uom: "cm" } } )
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
> db.inventory.find(  { size: { w: 21, h: 14, uom: "cm" } }  )
>
```

Query on Nested Field

To specify a query condition on fields in an embedded/nested document, use ***dot notation*** ("field.nestedField").

The following example selects all documents where the field *uom,* nested in the *size* field, equals "in":

```
> db.inventory.find()
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160aa"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
> db.inventory.find( { "size.uom": "in" } )
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
```

Specify Match using Query Operator

The following query uses the ***less than*** operator (*$lt*) on the field “h” embedded in the “size” field:

```
> db.inventory.find()
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160aa"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
> db.inventory.find( { "size.h": { $lt: 15 } } )
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
```

Specify AND Condition

The following query selects all documents where the ***nested field*** “h” is less than 15, the ***nested field*** “uom” equals "in", and the “status” field equals "D":

```
> db.inventory.find()
{ "_id" : ObjectId("5dade2b94a004f52120160a7"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a8"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160aa"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5dade2b94a004f52120160ab"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
> db.inventory.find( { "size.h": { $lt: 15 }, "size.uom": "in", status: "D" } )
{ "_id" : ObjectId("5dade2b94a004f52120160a9"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
```

Query an Array

Data to Load for the Example

```
db.inventory.insertMany([
   { item: "journal", qty: 25, tags: ["blank", "red"], dim_cm: [ 14, 21 ] },
   { item: "notebook", qty: 50, tags: ["red", "blank"], dim_cm: [ 14, 21 ] },
   { item: "paper", qty: 100, tags: ["red", "blank", "plain"], dim_cm: [ 14, 21 ] },
   { item: "planner", qty: 75, tags: ["blank", "red"], dim_cm: [ 22.85, 30 ] },
   { item: "postcard", qty: 45, tags: ["blue"], dim_cm: [ 10, 15.25 ] }
]);
```

Match an Array

To specify equality condition on an array, use the query document *{ <field>: <value> }* where <value> is the ***exact array to match***, including the ***order*** of the elements.

The following example queries for all documents where the ***field "tags" value*** is an array with exactly two elements, "red" and "blank", ***in the specified order***:

```
> db.inventory.find();
{ "_id" : ObjectId("5db6b2c20029e92d462c38d8"), "item" : "journal", "qty" : 25, "tags" : [ "blank", "red" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38d9"), "item" : "notebook", "qty" : 50, "tags" : [ "red", "blank" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38da"), "item" : "paper", "qty" : 100, "tags" : [ "red", "blank", "plain" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38db"), "item" : "planner", "qty" : 75, "tags" : [ "blank", "red" ], "dim_cm" : [ 22.85, 30 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38dc"), "item" : "postcard", "qty" : 45, "tags" : [ "blue" ], "dim_cm" : [ 10, 15.25 ] }
> db.inventory.find( { tags: ["red", "blank"] } )
{ "_id" : ObjectId("5db6b2c20029e92d462c38d9"), "item" : "notebook", "qty" : 50, "tags" : [ "red", "blank" ], "dim_cm" : [ 14, 21 ] }
> db.inventory.find( { tags: ["red", "blank"] } ).count();
1
```

Match an Array

If we wish to find an array that contains ***both the elements*** "red" and "blank", ***without regard to order*** or other elements in the array, use the ***$all*** operator:

```
> db.inventory.find();
{ "_id" : ObjectId("5db6b2c20029e92d462c38d8"), "item" : "journal", "qty" : 25, "tags" : [ "blank", "red" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38d9"), "item" : "notebook", "qty" : 50, "tags" : [ "red", "blank" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38da"), "item" : "paper", "qty" : 100, "tags" : [ "red", "blank", "plain" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38db"), "item" : "planner", "qty" : 75, "tags" : [ "blank", "red" ], "dim_cm" : [ 22.85, 30 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38dc"), "item" : "postcard", "qty" : 45, "tags" : [ "blue" ], "dim_cm" : [ 10, 15.25 ] }
> db.inventory.find( { tags: ["red", "blank"] } )
{ "_id" : ObjectId("5db6b2c20029e92d462c38d9"), "item" : "notebook", "qty" : 50, "tags" : [ "red", "blank" ], "dim_cm" : [ 14, 21 ] }
> db.inventory.find( { tags: { $all: ["red", "blank"] } } )
{ "_id" : ObjectId("5db6b2c20029e92d462c38d8"), "item" : "journal", "qty" : 25, "tags" : [ "blank", "red" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38d9"), "item" : "notebook", "qty" : 50, "tags" : [ "red", "blank" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38da"), "item" : "paper", "qty" : 100, "tags" : [ "red", "blank", "plain" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38db"), "item" : "planner", "qty" : 75, "tags" : [ "blank", "red" ], "dim_cm" : [ 22.85, 30 ] }
```

Query an Array for an Element

To query if the array field contains ***at least one*** element with the ***specified value***, use the filter { <field>: <value> } where <value> is the element value.

The following example queries for all documents where “tags“ is an array that ***contains the string "red" as one of its elements:***

```
> db.inventory.find( );
{ "_id" : ObjectId("5db99a948215ec9bc20c6c7f"), "item" : "journal", "qty" : 25, "tags" : [ "blank", "red" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db99a948215ec9bc20c6c80"), "item" : "notebook", "qty" : 50, "tags" : [ "red", "blank" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db99a948215ec9bc20c6c81"), "item" : "paper", "qty" : 100, "tags" : [ "red", "blank", "plain" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db99a948215ec9bc20c6c82"), "item" : "planner", "qty" : 75, "tags" : [ "blank", "red" ], "dim_cm" : [ 22.85, 30 ] }
{ "_id" : ObjectId("5db99a948215ec9bc20c6c83"), "item" : "postcard", "qty" : 45, "tags" : [ "blue" ], "dim_cm" : [ 10, 15.25 ] }
> db.inventory.find( { tags: "red" } )
{ "_id" : ObjectId("5db99a948215ec9bc20c6c7f"), "item" : "journal", "qty" : 25, "tags" : [ "blank", "red" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db99a948215ec9bc20c6c80"), "item" : "notebook", "qty" : 50, "tags" : [ "red", "blank" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db99a948215ec9bc20c6c81"), "item" : "paper", "qty" : 100, "tags" : [ "red", "blank", "plain" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db99a948215ec9bc20c6c82"), "item" : "planner", "qty" : 75, "tags" : [ "blank", "red" ], "dim_cm" : [ 22.85, 30 ] }
```

Specify Conditions in Array Elements

To specify conditions on the elements in the array field, use ***query operators*** in the ***query filter document*** { *<array field>: { <operator1>: <value1>, ... }* }

The following operations queries for all documents where the array ”dim_cm” contains at ***least one element*** whose value is greater than 25 (first query) and is greater than 20 (second query).

```
> db.inventory.find();
{ "_id" : ObjectId("5db6b2c20029e92d462c38d8"), "item" : "journal", "qty" : 25, "tags" : [ "blank", "red" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38d9"), "item" : "notebook", "qty" : 50, "tags" : [ "red", "blank" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38da"), "item" : "paper", "qty" : 100, "tags" : [ "red", "blank", "plain" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38db"), "item" : "planner", "qty" : 75, "tags" : [ "blank", "red" ], "dim_cm" : [ 22.85, 30 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38dc"), "item" : "postcard", "qty" : 45, "tags" : [ "blue" ], "dim_cm" : [ 10, 15.25 ] }
> db.inventory.find( { dim_cm: { $gt: 25 } } )
{ "_id" : ObjectId("5db6b2c20029e92d462c38db"), "item" : "planner", "qty" : 75, "tags" : [ "blank", "red" ], "dim_cm" : [ 22.85, 30 ] }
> db.inventory.find( { dim_cm: { $gt: 20 } } )
{ "_id" : ObjectId("5db6b2c20029e92d462c38d8"), "item" : "journal", "qty" : 25, "tags" : [ "blank", "red" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38d9"), "item" : "notebook", "qty" : 50, "tags" : [ "red", "blank" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38da"), "item" : "paper", "qty" : 100, "tags" : [ "red", "blank", "plain" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38db"), "item" : "planner", "qty" : 75, "tags" : [ "blank", "red" ], "dim_cm" : [ 22.85, 30 ] }
```

Multiple Conditions for Array Elements (I)

The following example, queries for documents where the “dim_cm” array contains elements that in ***some combination*** satisfy the query conditions; e.g., one element can satisfy the greater than 15 condition and another element can satisfy the less than 20 condition, or a single element can satisfy both:

```
> db.inventory.find();
{ "_id" : ObjectId("5db6b2c20029e92d462c38d8"), "item" : "journal", "qty" : 25, "tags" : [ "blank", "red" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38d9"), "item" : "notebook", "qty" : 50, "tags" : [ "red", "blank" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38da"), "item" : "paper", "qty" : 100, "tags" : [ "red", "blank", "plain" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38db"), "item" : "planner", "qty" : 75, "tags" : [ "blank", "red" ], "dim_cm" : [ 22.85, 30 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38dc"), "item" : "postcard", "qty" : 45, "tags" : [ "blue" ], "dim_cm" : [ 10, 15.25 ] }
> db.inventory.find( { dim_cm: { $gt: 15, $lt: 20 } } )
{ "_id" : ObjectId("5db6b2c20029e92d462c38d8"), "item" : "journal", "qty" : 25, "tags" : [ "blank", "red" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38d9"), "item" : "notebook", "qty" : 50, "tags" : [ "red", "blank" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38da"), "item" : "paper", "qty" : 100, "tags" : [ "red", "blank", "plain" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38dc"), "item" : "postcard", "qty" : 45, "tags" : [ "blue" ], "dim_cm" : [ 10, 15.25 ] }
```

Has been excluded because none of the elements satisfies the condition $lt:20

Multiple Conditions for Array Elements (II)

The following example queries for documents where the “dim_cm” array contains ***at least one element that is both*** greater than ($gt) 22 and less than ($lt) 30:

```
> db.inventory.find();
{ "_id" : ObjectId("5db6b2c20029e92d462c38d8"), "item" : "journal", "qty" : 25, "tags" : [ "blank", "red" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38d9"), "item" : "notebook", "qty" : 50, "tags" : [ "red", "blank" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38da"), "item" : "paper", "qty" : 100, "tags" : [ "red", "blank", "plain" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38db"), "item" : "planner", "qty" : 75, "tags" : [ "blank", "red" ], "dim_cm" : [ 22.85, 30 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38dc"), "item" : "postcard", "qty" : 45, "tags" : [ "blue" ], "dim_cm" : [ 10, 15.25 ] }
> db.inventory.find( { dim_cm: { $elemMatch: { $gt: 22, $lt: 30 } } } )
{ "_id" : ObjectId("5db6b2c20029e92d462c38db"), "item" : "planner", "qty" : 75, "tags" : [ "blank", "red" ], "dim_cm" : [ 22.85, 30 ] }
>
```

Query for an Element by the Array Index Position

Using ***dot notation***, we can specify query conditions for an element at a particular index or position of the array. The array uses ***zero-based indexing***.

The following example queries for all documents where ***the second element*** in the array "dim_cm" ***is greater than 25***:

```
> db.inventory.find();
{ "_id" : ObjectId("5db6b2c20029e92d462c38d8"), "item" : "journal", "qty" : 25, "tags" : [ "blank", "red" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38d9"), "item" : "notebook", "qty" : 50, "tags" : [ "red", "blank" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38da"), "item" : "paper", "qty" : 100, "tags" : [ "red", "blank", "plain" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38db"), "item" : "planner", "qty" : 75, "tags" : [ "blank", "red" ], "dim_cm" : [ 22.85, 30 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38dc"), "item" : "postcard", "qty" : 45, "tags" : [ "blue" ], "dim_cm" : [ 10, 15.25 ] }
> db.inventory.find( { "dim_cm.1": { $gt: 25 } } )
{ "_id" : ObjectId("5db6b2c20029e92d462c38db"), "item" : "planner", "qty" : 75, "tags" : [ "blank", "red" ], "dim_cm" : [ 22.85, 30 ] }
```

Query an Array by Array Length

Use the *$size* operator to query for arrays by number of elements.

For example, the following selects documents where the ***array "tags" has 3 elements***.

```
> db.inventory.find();
{ "_id" : ObjectId("5db6b2c20029e92d462c38d8"), "item" : "journal", "qty" : 25, "tags" : [ "blank", "red" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38d9"), "item" : "notebook", "qty" : 50, "tags" : [ "red", "blank" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38da"), "item" : "paper", "qty" : 100, "tags" : [ "red", "blank", "plain" ], "dim_cm" : [ 14, 21 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38db"), "item" : "planner", "qty" : 75, "tags" : [ "blank", "red" ], "dim_cm" : [ 22.85, 30 ] }
{ "_id" : ObjectId("5db6b2c20029e92d462c38dc"), "item" : "postcard", "qty" : 45, "tags" : [ "blue" ], "dim_cm" : [ 10, 15.25 ] }
> db.inventory.find( { "tags": { $size: 3 } } )
{ "_id" : ObjectId("5db6b2c20029e92d462c38da"), "item" : "paper", "qty" : 100, "tags" : [ "red", "blank", "plain" ], "dim_cm" : [ 14, 21 ] }
```

Suggested Readings

Students are invited to read the official documentation regarding cursors operations with MongoDB.

The documentation is available at:
https://docs.mongodb.com/manual/reference/method/db.collection.find/
https://docs.mongodb.com/manual/reference/method/db.collection.countDocuments/
https://docs.mongodb.com/manual/tutorial/query-embedded-documents/
https://docs.mongodb.com/manual/tutorial/query-arrays/

Students are also invited to repeat all the examples on their MongoDB shell.

Exercises

1. Import the restaurants datasets from the restaurants.json file.
2. Write a MongoDB query to count the total number of documents.
3. Write a MongoDB query to display all the restaurant which is in the borough Bronx
4. Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx
5. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100
6. Write a MongoDB query to find the restaurants which locate in a latitude value less than -95.754168.
7. Write a MongoDB query to find the restaurants that do not prepare any cuisine of 'American ' and their grade score more than 70 and latitude less than -65.754168
8. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American ' and achieved a grade point 'A' not belongs to the borough Brooklyn. The document must be displayed according to the cuisine in descending order.