MongoDB Practice Questions

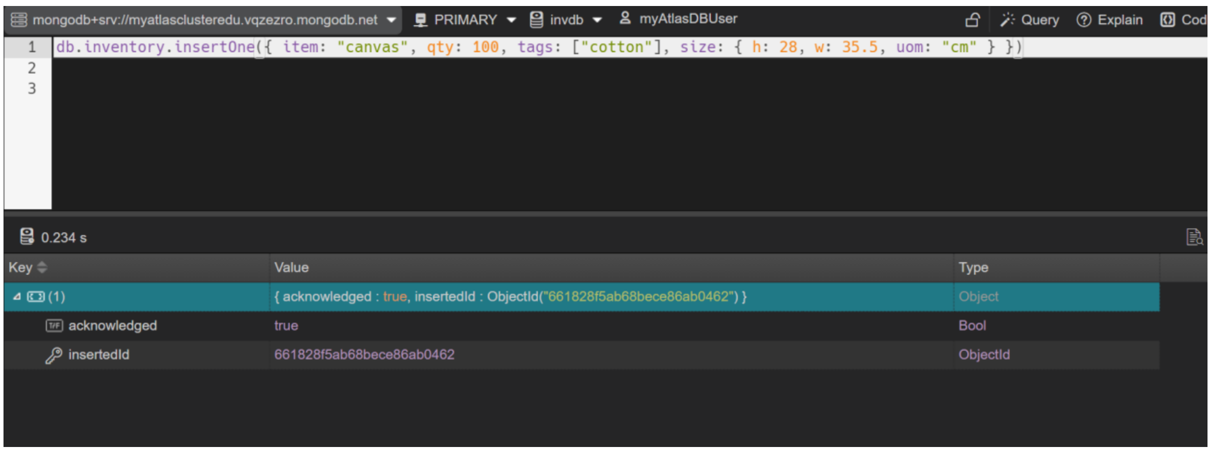
CRUD Operations

Create

insertOne

Create collection inventory and insert Inventory details

db.inventory.insertOne({ item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom: "cm" }})



```
db.inventory.insertOne({ item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom: "cm" } })
```

```
0.234 s
Key               Value                                                                        Type
(1)               { acknowledged : true, insertedId : ObjectId("661828f5ab68bece86ab0462") }   Object
  acknowledged    true                                                                         Bool
  insertedId      661828f5ab68bece86ab0462                                                     ObjectId
```

insertMany

db.inventory.insertMany([

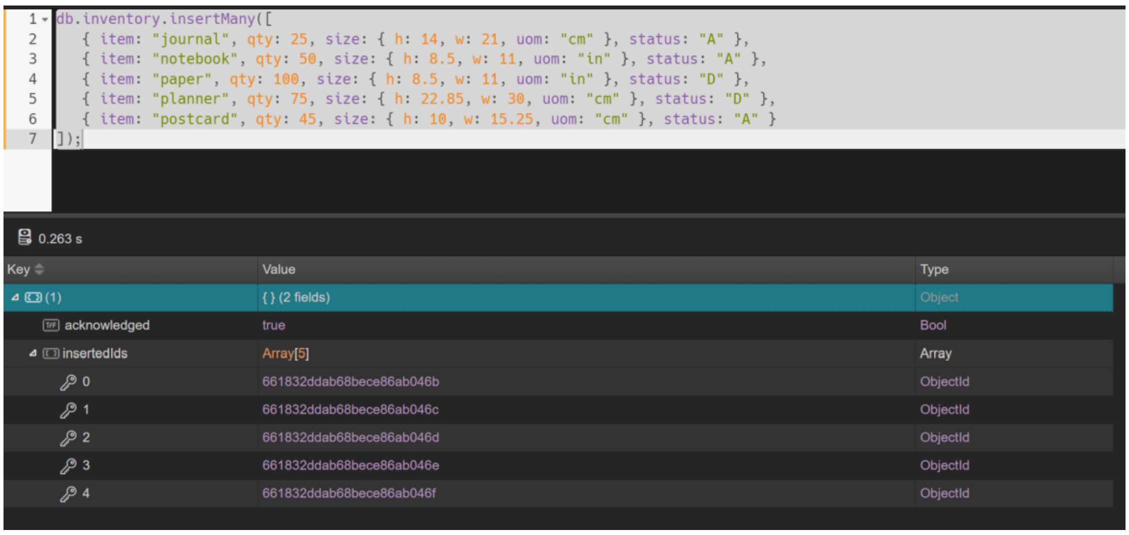
{ item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },

{ 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" }]);



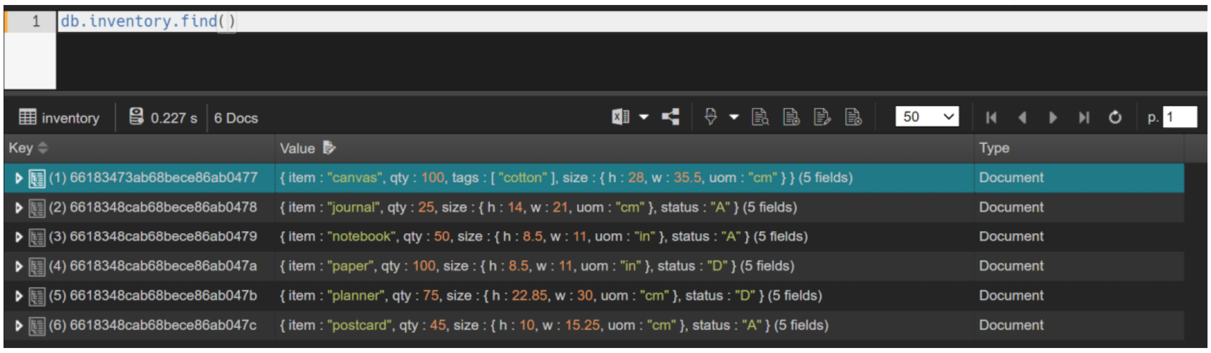
```
db.inventory.insertMany([
   { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
   { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },
   { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
   { item: "planner", qty: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

```
0.263 s
Key               Value                          Type
(1)               { } (2 fields)                 Object
  acknowledged    true                           Bool
  insertedIds     Array[5]                       Array
    0             661832ddab68bece86ab046b       ObjectId
    1             661832ddab68bece86ab046c       ObjectId
    2             661832ddab68bece86ab046d       ObjectId
    3             661832ddab68bece86ab046e       ObjectId
    4             661832ddab68bece86ab046f       ObjectId
```

Read

find()

db.inventory.find()

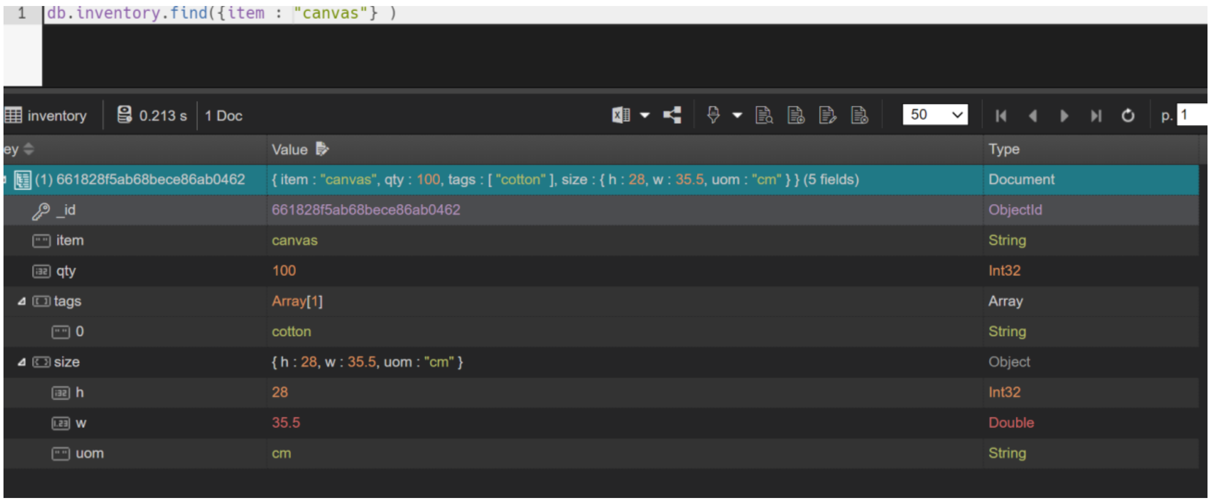


```
db.inventory.find()
```

```
inventory   0.227 s   6 Docs                                                        50   p. 1
Key                              Value                                                                                              Type
(1) 66183473ab68bece86ab0477     { item : "canvas", qty : 100, tags : [ "cotton" ], size : { h : 28, w : 35.5, uom : "cm" } } (5 fields)   Document
(2) 6618348cab68bece86ab0478     { item : "journal", qty : 25, size : { h : 14, w : 21, uom : "cm" }, status : "A" } (5 fields)            Document
(3) 6618348cab68bece86ab0479     { item : "notebook", qty : 50, size : { h : 8.5, w : 11, uom : "in" }, status : "A" } (5 fields)          Document
(4) 6618348cab68bece86ab047a     { item : "paper", qty : 100, size : { h : 8.5, w : 11, uom : "in" }, status : "D" } (5 fields)            Document
(5) 6618348cab68bece86ab047b     { item : "planner", qty : 75, size : { h : 22.85, w : 30, uom : "cm" }, status : "D" } (5 fields)         Document
(6) 6618348cab68bece86ab047c     { item : "postcard", qty : 45, size : { h : 10, w : 15.25, uom : "cm" }, status : "A" } (5 fields)        Document
```

To specify equality conditions, use `<field>:<value>` expressions in the query filter document

db.inventory.find({item : "canvas"})



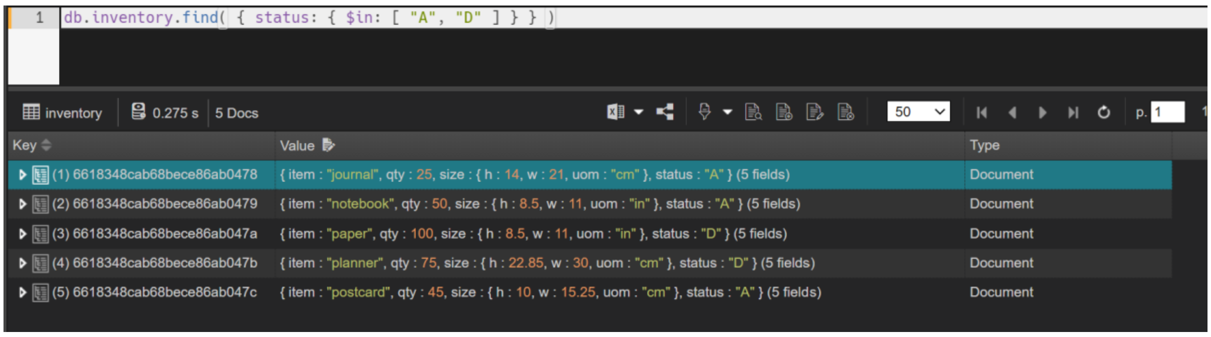
```
db.inventory.find({item : "canvas"} )
```

```
inventory   0.213 s   1 Doc                                                         50   p. 1
ey                               Value                                                                                              Type
(1) 661828f5ab68bece86ab0462     { item : "canvas", qty : 100, tags : [ "cotton" ], size : { h : 28, w : 35.5, uom : "cm" } } (5 fields)   Document
    _id                          661828f5ab68bece86ab0462                                                                           ObjectId
    item                         canvas                                                                                             String
    qty                          100                                                                                                Int32
    tags                         Array[1]                                                                                           Array
        0                        cotton                                                                                             String
    size                         { h : 28, w : 35.5, uom : "cm" }                                                                   Object
        h                        28                                                                                                 Int32
        w                        35.5                                                                                               Double
        uom                      cm                                                                                                 String
```

A query filter documents can use the query operators to specify conditions

$in

db.inventory.find({ status: { $in: ["A", "D"] } })



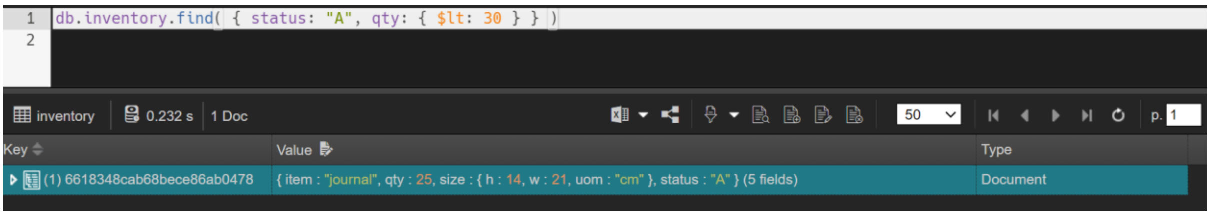
```
db.inventory.find( { status: { $in: [ "A", "D" ] } } )
```

```
inventory   0.275 s   5 Docs                                                        50   p. 1
Key                              Value                                                                                              Type
(1) 6618348cab68bece86ab0478     { item : "journal", qty : 25, size : { h : 14, w : 21, uom : "cm" }, status : "A" } (5 fields)            Document
(2) 6618348cab68bece86ab0479     { item : "notebook", qty : 50, size : { h : 8.5, w : 11, uom : "in" }, status : "A" } (5 fields)          Document
(3) 6618348cab68bece86ab047a     { item : "paper", qty : 100, size : { h : 8.5, w : 11, uom : "in" }, status : "D" } (5 fields)            Document
(4) 6618348cab68bece86ab047b     { item : "planner", qty : 75, size : { h : 22.85, w : 30, uom : "cm" }, status : "D" } (5 fields)         Document
(5) 6618348cab68bece86ab047c     { item : "postcard", qty : 45, size : { h : 10, w : 15.25, uom : "cm" }, status : "A" } (5 fields)        Document
```

A compound query can specify conditions for more than one field in the collection's documents

AND

db.inventory.find({ status: "A", qty: { $lt: 30 } })

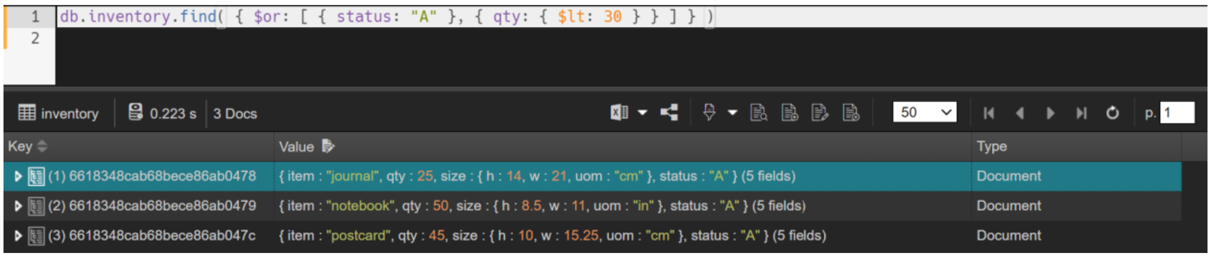


```
db.inventory.find( { status: "A", qty: { $lt: 30 } } )
```

```
inventory   0.232 s   1 Doc
Key                                Value                                                                                   Type
(1) 6618348cab68bece86ab0478       { item : "journal", qty : 25, size : { h : 14, w : 21, uom : "cm" }, status : "A" } (5 fields)   Document
```

OR

db.inventory.find({ $or: [{ status: "A" }, { qty: { $lt: 30 } }] })



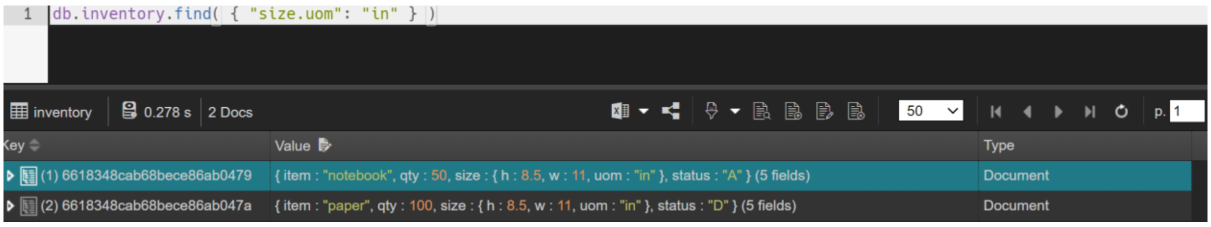
```
db.inventory.find( { $or: [ { status: "A" }, { qty: { $lt: 30 } } ] } )
```

```
inventory   0.223 s   3 Docs
Key                                Value                                                                                          Type
(1) 6618348cab68bece86ab0478       { item : "journal", qty : 25, size : { h : 14, w : 21, uom : "cm" }, status : "A" } (5 fields)      Document
(2) 6618348cab68bece86ab0479       { item : "notebook", qty : 50, size : { h : 8.5, w : 11, uom : "in" }, status : "A" } (5 fields)    Document
(3) 6618348cab68bece86ab047c       { item : "postcard", qty : 45, size : { h : 10, w : 15.25, uom : "cm" }, status : "A" } (5 fields)  Document
```

Query on Embedded documents/NestedFields

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

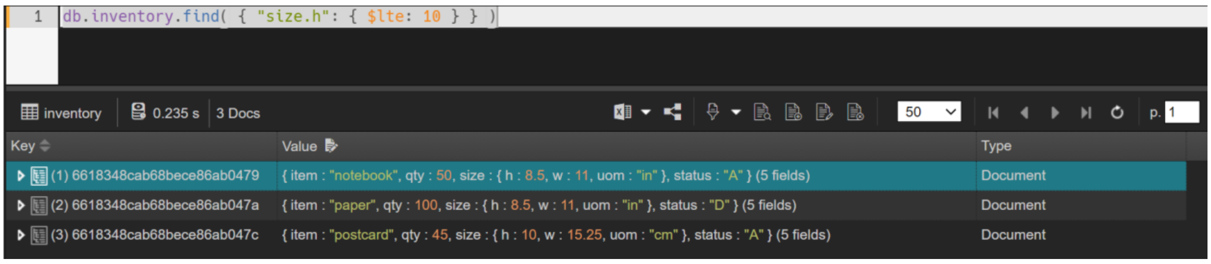
db.inventory.find({ "size.uom": "in" })



```
db.inventory.find( { "size.uom": "in" } )
```

```
inventory   0.278 s   2 Docs
Key                                Value                                                                                         Type
(1) 6618348cab68bece86ab0479       { item : "notebook", qty : 50, size : { h : 8.5, w : 11, uom : "in" }, status : "A" } (5 fields)   Document
(2) 6618348cab68bece86ab047a       { item : "paper", qty : 100, size : { h : 8.5, w : 11, uom : "in" }, status : "D" } (5 fields)     Document
```

db.inventory.find({ "size.h": { $lte: 10 } })



```
db.inventory.find( { "size.h": { $lte: 10 } } )
```

```
inventory   0.235 s   3 Docs
Key                                Value                                                                                          Type
(1) 6618348cab68bece86ab0479       { item : "notebook", qty : 50, size : { h : 8.5, w : 11, uom : "in" }, status : "A" } (5 fields)    Document
(2) 6618348cab68bece86ab047a       { item : "paper", qty : 100, size : { h : 8.5, w : 11, uom : "in" }, status : "D" } (5 fields)      Document
(3) 6618348cab68bece86ab047c       { item : "postcard", qty : 45, size : { h : 10, w : 15.25, uom : "cm" }, status : "A" } (5 fields)  Document
```

Query on Arrays

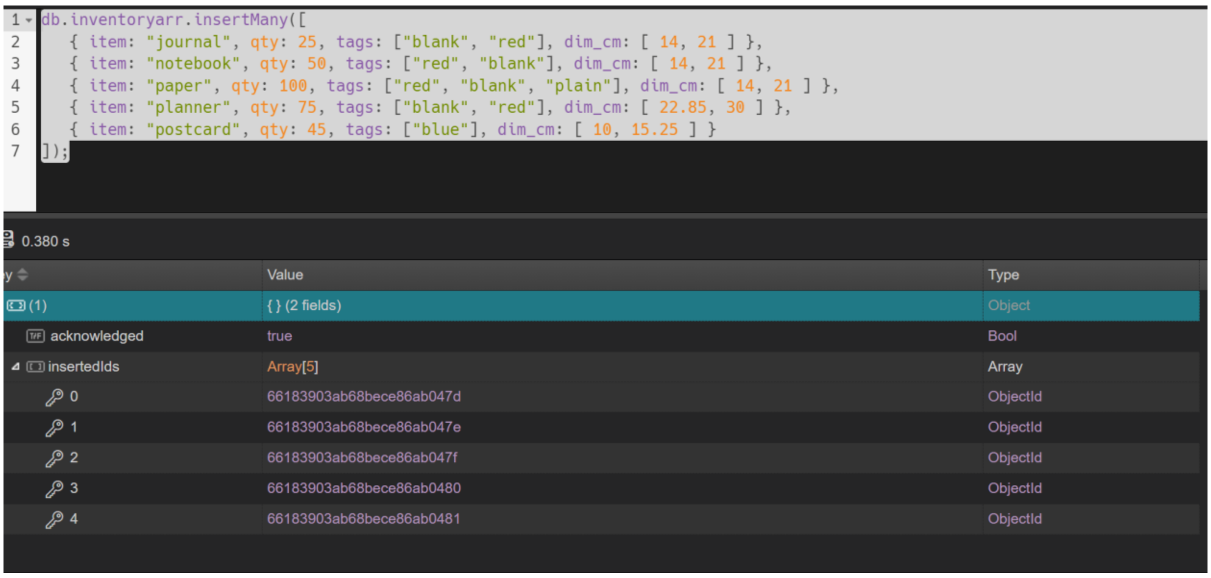
Inventory collection with array

```
db.inventoryarr.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] }

]);



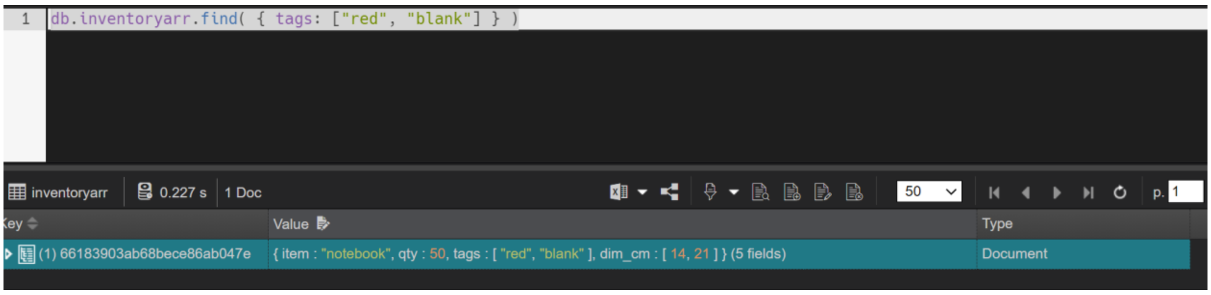
```
db.inventoryarr.insertMany([
   { item: "journal", qty: 25, tags: ["blank", "red"], dim_cm: [ 14, 21 ] },
   { item: "notebook", qty: 50, tags: ["red", "blank"], dim_cm: [ 14, 21 ] },
   { item: "paper", qty: 100, tags: ["red", "blank", "plain"], dim_cm: [ 14, 21 ] },
   { item: "planner", qty: 75,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

```
0.380 s
y                  Value                        Type
(1)                { } (2 fields)               Object
  acknowledged     true                         Bool
  insertedIds      Array[5]                     Array
    0              66183903ab68bece86ab047d     ObjectId
    1              66183903ab68bece86ab047e     ObjectId
    2              66183903ab68bece86ab047f     ObjectId
    3              66183903ab68bece86ab0480     ObjectId
    4              66183903ab68bece86ab0481     ObjectId
```

To specify equality condition on an array

Queries for all documents where the field `tags` value is an array with exactly two elements, `"red"` and `"blank"`, in the specified order

db.inventoryarr.find({ tags: ["red", "blank"] })

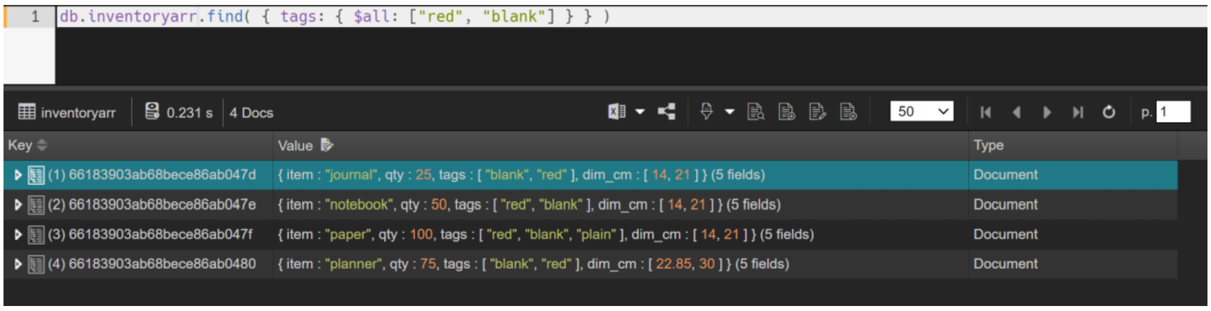


```
db.inventoryarr.find( { tags: ["red", "blank"] } )
```

```
inventoryarr   0.227 s   1 Doc                                        50   p. 1
Key                                  Value                                                                              Type
(1) 66183903ab68bece86ab047e         { item : "notebook", qty : 50, tags : [ "red", "blank" ], dim_cm : [ 14, 21 ] } (5 fields)   Document
```

Query 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.inventoryarr.find({ tags: { $all: ["red", "blank"] } })



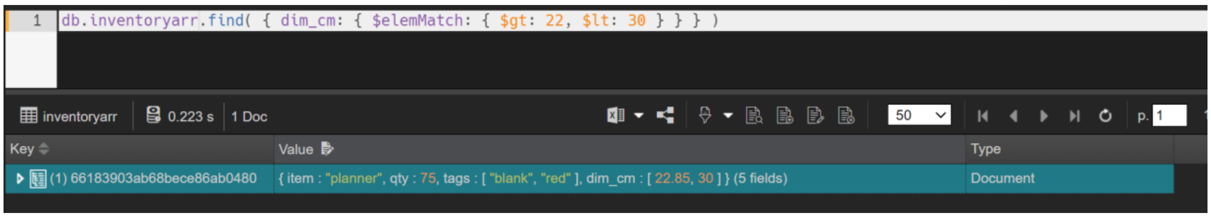
```
db.inventoryarr.find( { tags: { $all: ["red", "blank"] } } )
```

```
inventoryarr   0.231 s   4 Docs                                       50   p. 1
Key                                  Value                                                                                        Type
(1) 66183903ab68bece86ab047d         { item : "journal", qty : 25, tags : [ "blank", "red" ], dim_cm : [ 14, 21 ] } (5 fields)            Document
(2) 66183903ab68bece86ab047e         { item : "notebook", qty : 50, tags : [ "red", "blank" ], dim_cm : [ 14, 21 ] } (5 fields)           Document
(3) 66183903ab68bece86ab047f         { item : "paper", qty : 100, tags : [ "red", "blank", "plain" ], dim_cm : [ 14, 21 ] } (5 fields)    Document
(4) 66183903ab68bece86ab0480         { item : "planner", qty : 75, tags : [ "blank", "red" ], dim_cm : [ 22.85, 30 ] } (5 fields)         Document
```

Query an array elements that meets multiple criteria

The $elemMatch operator matches documents that contain an array field with at least one element that matches all the specified query criteria

db.inventoryarr.find({ dim_cm: { $elemMatch: { $gt: 22, $lt: 30 } } })

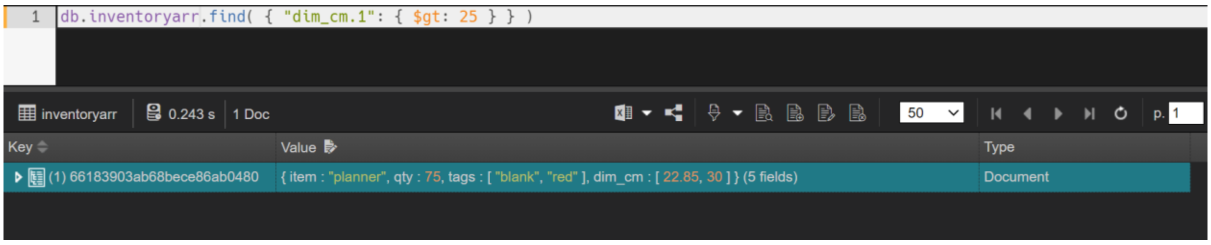


```
db.inventoryarr.find( { dim_cm: { $elemMatch: { $gt: 22, $lt: 30 } } } )
```

```
inventoryarr   0.223 s   1 Doc                                   50   p. 1
Key                                Value                                                                                   Type
(1) 66183903ab68bece86ab0480       { item : "planner", qty : 75, tags : [ "blank", "red" ], dim_cm : [ 22.85, 30 ] } (5 fields)   Document
```

Query for an Element by the Array Index Position

db.inventoryarr.find({ "dim_cm.1": { $gt: 25 } })

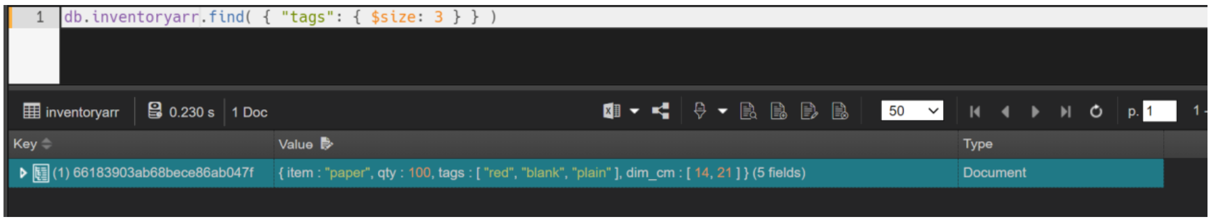


```
db.inventoryarr.find( { "dim_cm.1": { $gt: 25 } } )
```

```
inventoryarr   0.243 s   1 Doc                                   50   p. 1
Key                                Value                                                                                   Type
(1) 66183903ab68bece86ab0480       { item : "planner", qty : 75, tags : [ "blank", "red" ], dim_cm : [ 22.85, 30 ] } (5 fields)   Document
```

Query an Array by Array Length

db.inventoryarr.find({ "tags": { $size: 3 } })



```
db.inventoryarr.find( { "tags": { $size: 3 } } )
```

```
inventoryarr   0.230 s   1 Doc                                   50   p. 1
Key                                Value                                                                                   Type
(1) 66183903ab68bece86ab047f       { item : "paper", qty : 100, tags : [ "red", "blank", "plain" ], dim_cm : [ 14, 21 ] } (5 fields)   Document
```

Query an Array on Embedded documents

db.inventoryembed.insertMany([

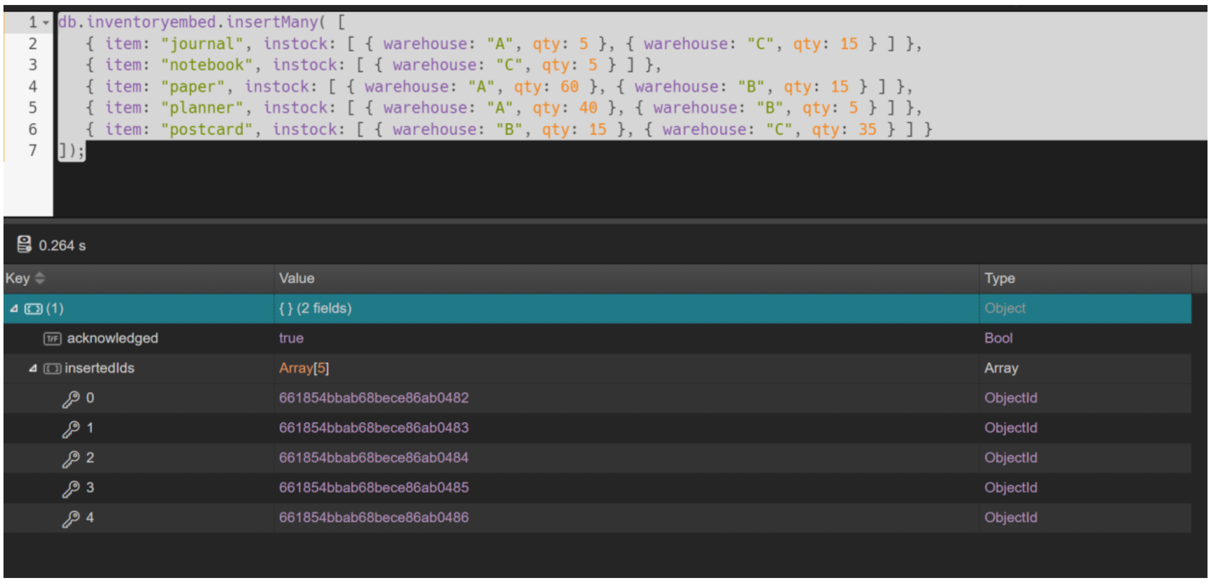
{ item: "journal", instock: [{ warehouse: "A", qty: 5 }, { warehouse: "C", qty: 15 }] },

{ item: "notebook", instock: [{ warehouse: "C", qty: 5 }] },

{ item: "paper", instock: [{ warehouse: "A", qty: 60 }, { warehouse: "B", qty: 15 }] },

{ item: "planner", instock: [{ warehouse: "A", qty: 40 }, { warehouse: "B", qty: 5 }] },

{ item: "postcard", instock: [{ warehouse: "B", qty: 15 }, { warehouse: "C", qty: 35 }] }]);

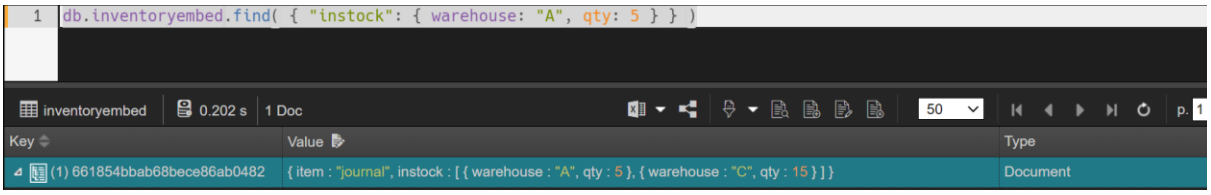


```
db.inventoryembed.insertMany( [
   { item: "journal", instock: [ { warehouse: "A", qty: 5 }, { warehouse: "C", qty: 15 } ] },
   { item: "notebook", instock: [ { warehouse: "C", qty: 5 } ] },
   { item: "paper", instock: [ { warehouse: "A", qty: 60 }, { warehouse: "B", qty: 15 } ] },
   { item: "planner", instock: [ { warehouse: "A", qty: 40 }, { warehouse: "B", qty: 5 } ] },
   { item: "postcard", instock: [ { warehouse: "B", qty: 15 }, { warehouse: "C", qty: 35 } ] }
]);
```

```
0.264 s
Key                 Value                        Type
(1)                 { } (2 fields)               Object
  acknowledged      true                         Bool
  insertedIds       Array[5]                     Array
    0               661854bbab68bece86ab0482     ObjectId
    1               661854bbab68bece86ab0483     ObjectId
    2               661854bbab68bece86ab0484     ObjectId
    3               661854bbab68bece86ab0485     ObjectId
    4               661854bbab68bece86ab0486     ObjectId
```

Query

db.inventoryembed.find({ "instock": { warehouse: "A", qty: 5 } })

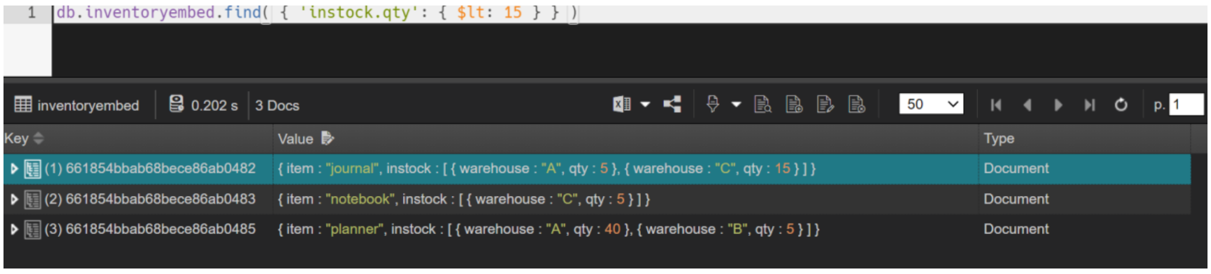


```
db.inventoryembed.find( { "instock": { warehouse: "A", qty: 5 } } )
```

```
inventoryembed   0.202 s   1 Doc
Key                                  Value                                                                                   Type
(1) 661854bbab68bece86ab0482         { item : "journal", instock : [ { warehouse : "A", qty : 5 }, { warehouse : "C", qty : 15 } ] }   Document
```

Specify a Query Condition on a Field Embedded in an Array of Documents

db.inventoryembed.find({ 'instock.qty': { $lt: 15 } })

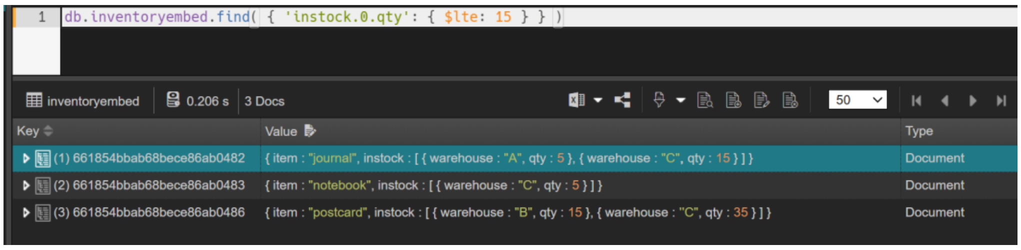


```
db.inventoryembed.find( { 'instock.qty': { $lt: 15 } } )
```

```
inventoryembed   0.202 s   3 Docs
Key                                  Value                                                                                   Type
(1) 661854bbab68bece86ab0482         { item : "journal", instock : [ { warehouse : "A", qty : 5 }, { warehouse : "C", qty : 15 } ] }   Document
(2) 661854bbab68bece86ab0483         { item : "notebook", instock : [ { warehouse : "C", qty : 5 } ] }                                  Document
(3) 661854bbab68bece86ab0485         { item : "planner", instock : [ { warehouse : "A", qty : 40 }, { warehouse : "B", qty : 5 } ] }   Document
```

Use the Array Index to Query for a Field in the Embedded Document

db.inventoryembed.find({ 'instock.0.qty': { $lte: 15 } })



```
db.inventoryembed.find( { 'instock.0.qty': { $lte: 15 } } )
```

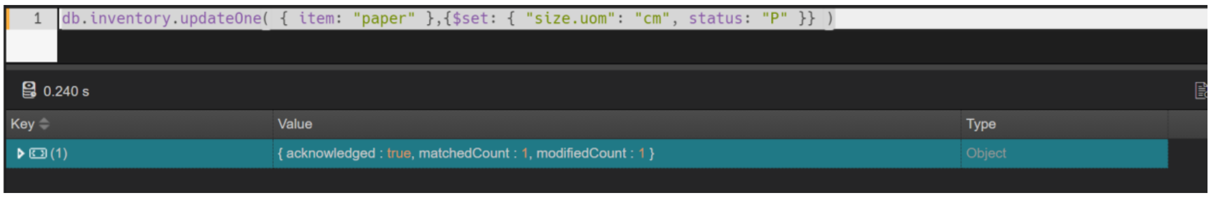
```
inventoryembed   0.206 s   3 Docs
Key                                  Value                                                                                   Type
(1) 661854bbab68bece86ab0482         { item : "journal", instock : [ { warehouse : "A", qty : 5 }, { warehouse : "C", qty : 15 } ] }   Document
(2) 661854bbab68bece86ab0483         { item : "notebook", instock : [ { warehouse : "C", qty : 5 } ] }                                  Document
(3) 661854bbab68bece86ab0486         { item : "postcard", instock : [ { warehouse : "B", qty : 15 }, { warehouse : "C", qty : 35 } ] }  Document
```

Update

MongoDB provides the updateOne() or updateMany() methods to perform update operations.

updateOne

db.inventory.updateOne({ item: "paper" },{$set: { "size.uom": "cm", status: "P" }})

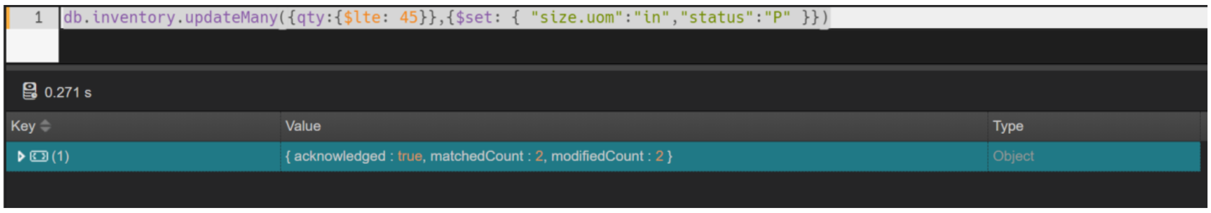


```
db.inventory.updateOne( { item: "paper" },{$set: { "size.uom": "cm", status: "P" }} )
```

```
0.240 s
Key        Value                                                     Type
(1)        { acknowledged : true, matchedCount : 1, modifiedCount : 1 }   Object
```

updateMany

db.inventory.updateMany({qty:{$lte: 45}},{$set: { "size.uom":"in","status":"P" }})

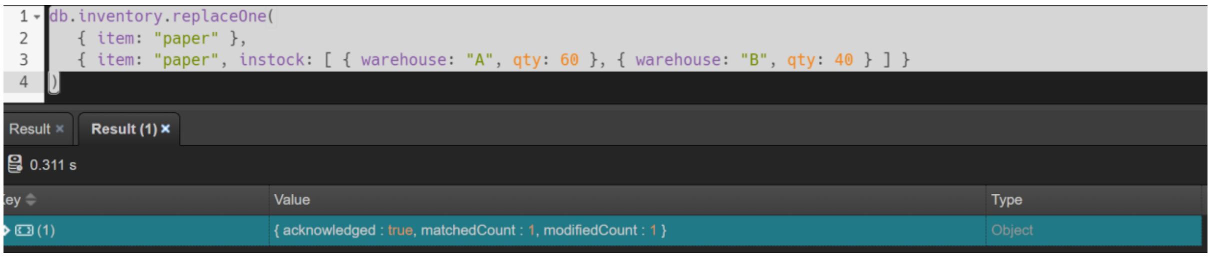


```
db.inventory.updateMany({qty:{$lte: 45}},{$set: { "size.uom":"in","status":"P" }})
```

```
0.271 s
Key        Value                                                     Type
(1)        { acknowledged : true, matchedCount : 2, modifiedCount : 2 }   Object
```

To replace the entire content of a document except for the _id field, pass an entirely new document as the second argument to db.collection.replaceOne()

db.inventory.replaceOne({ item: "paper" }, { item: "paper", instock: [{ warehouse: "A", qty: 60 }, { warehouse: "B", qty: 40 }] })



```
db.inventory.replaceOne(
   { item: "paper" },
   { item: "paper", instock: [ { warehouse: "A", qty: 60 }, { warehouse: "B", qty: 40 } ] }
)
```

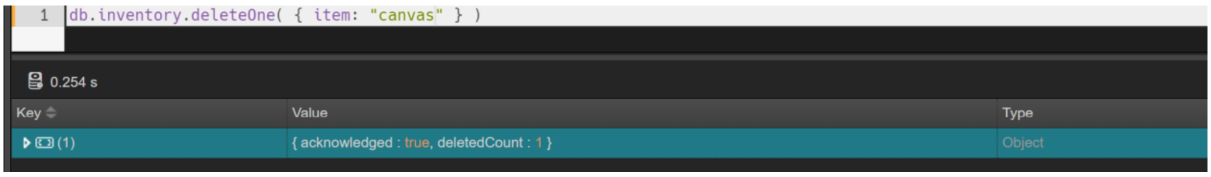
```
0.311 s
Key        Value                                                     Type
(1)        { acknowledged : true, matchedCount : 1, modifiedCount : 1 }   Object
```

Delete

Deleting documents in MongoDB is done using the deleteOne() or deleteMany() methods.

deleteOne

db.inventory.deleteOne({ item: "canvas" })

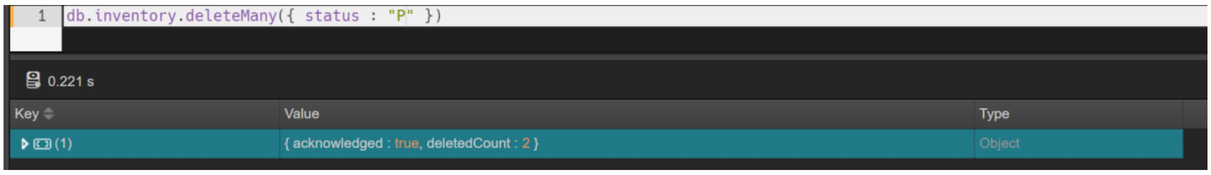


```
db.inventory.deleteOne( { item: "canvas" } )
```

```
0.254 s
Key        Value                                      Type
(1)        { acknowledged : true, deletedCount : 1 }   Object
```

deleteMany

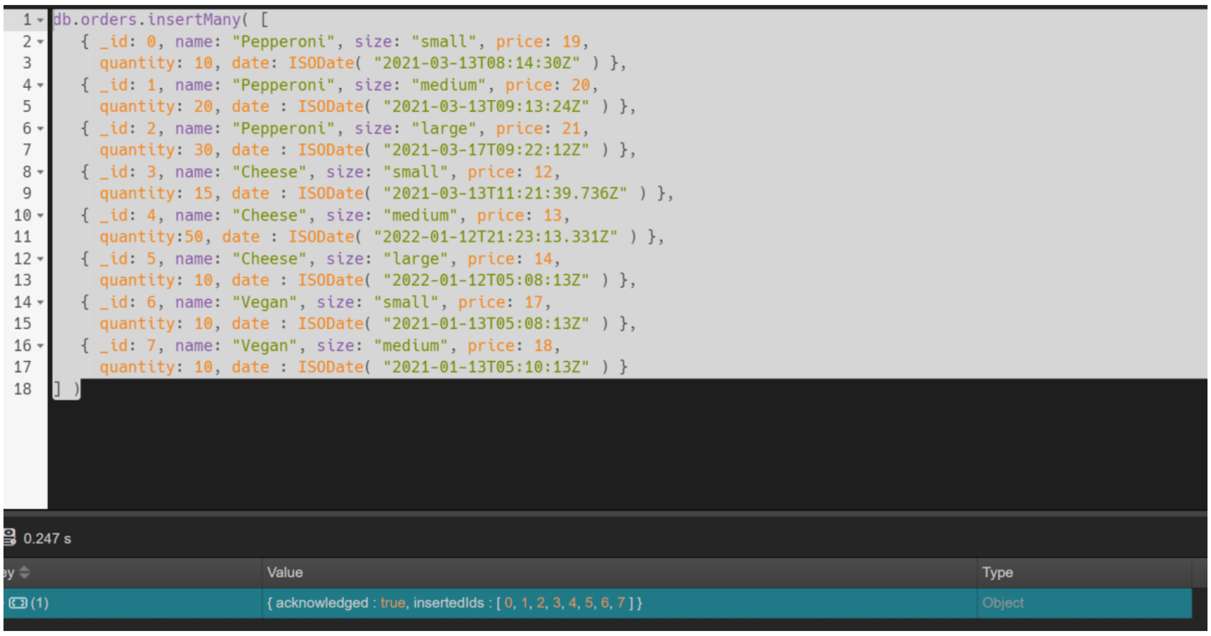
db.inventory.deleteMany({ status : "P" })



```
db.inventory.deleteMany({ status : "P" })
```

```
0.221 s
Key        Value                                              Type
(1)        { acknowledged : true, deletedCount : 2 }          Object
```

Aggregation Pipelines



```
db.orders.insertMany( [
   { _id: 0, name: "Pepperoni", size: "small", price: 19,
     quantity: 10, date: ISODate( "2021-03-13T08:14:30Z" ) },
   { _id: 1, name: "Pepperoni", size: "medium", price: 20,
     quantity: 20, date : ISODate( "2021-03-13T09:13:24Z" ) },
   { _id: 2, name: "Pepperoni", size: "large", price: 21,
     quantity: 30, date : ISODate( "2021-03-17T09:22:12Z" ) },
   { _id: 3, name: "Cheese", size: "small", price: 12,
     quantity: 15, date : ISODate( "2021-03-13T11:21:39.736Z" ) },
   { _id: 4, name: "Cheese", size: "medium", price: 13,
     quantity:50, date : ISODate( "2022-01-12T21:23:13.331Z" ) },
   { _id: 5, name: "Cheese", size: "large", price: 14,
     quantity: 10, date : ISODate( "2022-01-12T05:08:13Z" ) },
   { _id: 6, name: "Vegan", size: "small", price: 17,
     quantity: 10, date : ISODate( "2021-01-13T05:08:13Z" ) },
   { _id: 7, name: "Vegan", size: "medium", price: 18,
     quantity: 10, date : ISODate( "2021-01-13T05:10:13Z" ) }
] )
```

```
0.247 s
ey         Value                                                          Type
(1)        { acknowledged : true, insertedIds : [ 0, 1, 2, 3, 4, 5, 6, 7 ] }   Object
```

Calculate total order quantity of medium size pizzas grouped by pizza name

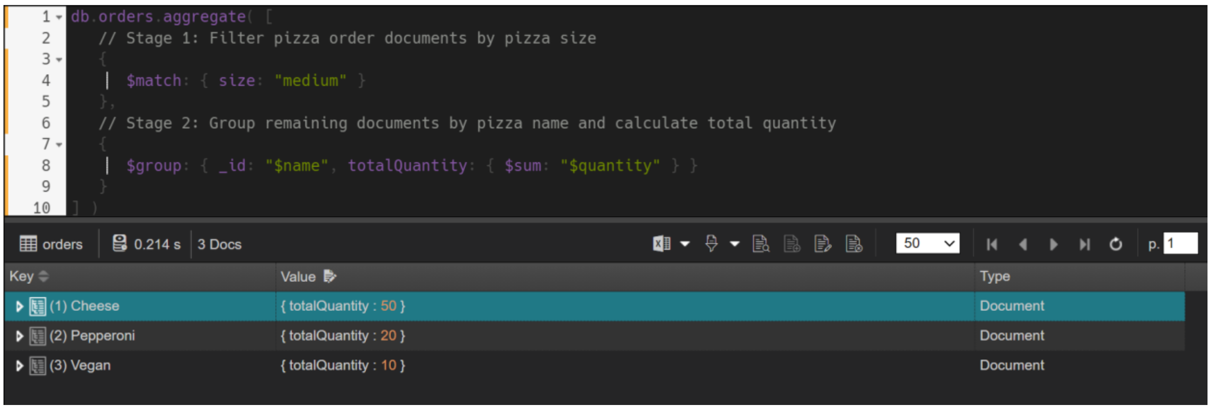
db.orders.aggregate([

// Stage 1: Filter pizza order documents by pizza size

{$match: { size: "medium" } },

// Stage 2: Group remaining documents by pizza name and calculate total quantity

{ $group: { _id: "$name", totalQuantity: { $sum: "$quantity" } }}])



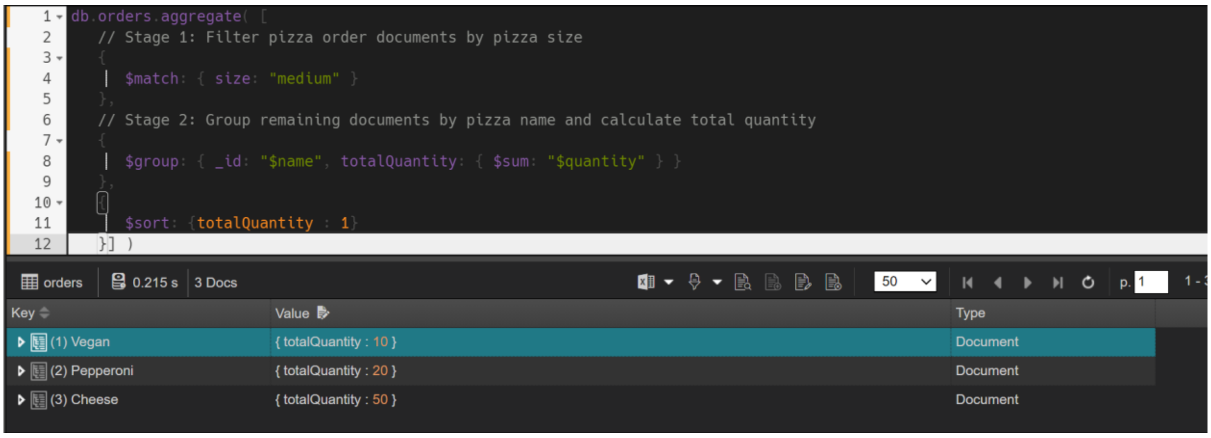
```
db.orders.aggregate( [
   // Stage 1: Filter pizza order documents by pizza size
   {
     | $match: { size: "medium" }
   },
   // Stage 2: Group remaining documents by pizza name and calculate total quantity
   {
     | $group: { _id: "$name", totalQuantity: { $sum: "$quantity" } }
   }
] )
```

```
orders   0.214 s   3 Docs                                   50   p. 1
Key              Value                           Type
(1) Cheese       { totalQuantity : 50 }          Document
(2) Pepperoni    { totalQuantity : 20 }          Document
(3) Vegan        { totalQuantity : 10 }          Document
```

db.orders.aggregate([

// Stage 1: Filter pizza order documents by pizza size

```
  { $match: { size: "medium" }  },

  // Stage 2: Group remaining documents by pizza name and calculate total quantity

  { $group: { _id: "$name", totalQuantity: { $sum: "$quantity" } }},

  {     $sort: {totalQuantity : 1}     }] )
```



```
db.orders.aggregate( [
   // Stage 1: Filter pizza order documents by pizza size
   {
   |  $match: { size: "medium" }
   },
   // Stage 2: Group remaining documents by pizza name and calculate total quantity
   {
   |  $group: { _id: "$name", totalQuantity: { $sum: "$quantity" } }
   },
   {
   |  $sort: {totalQuantity : 1}
   }] )
```

```
orders   0.215 s   3 Docs

Key                 Value                        Type
(1) Vegan           { totalQuantity : 10 }       Document
(2) Pepperoni       { totalQuantity : 20 }       Document
(3) Cheese          { totalQuantity : 50 }       Document
```

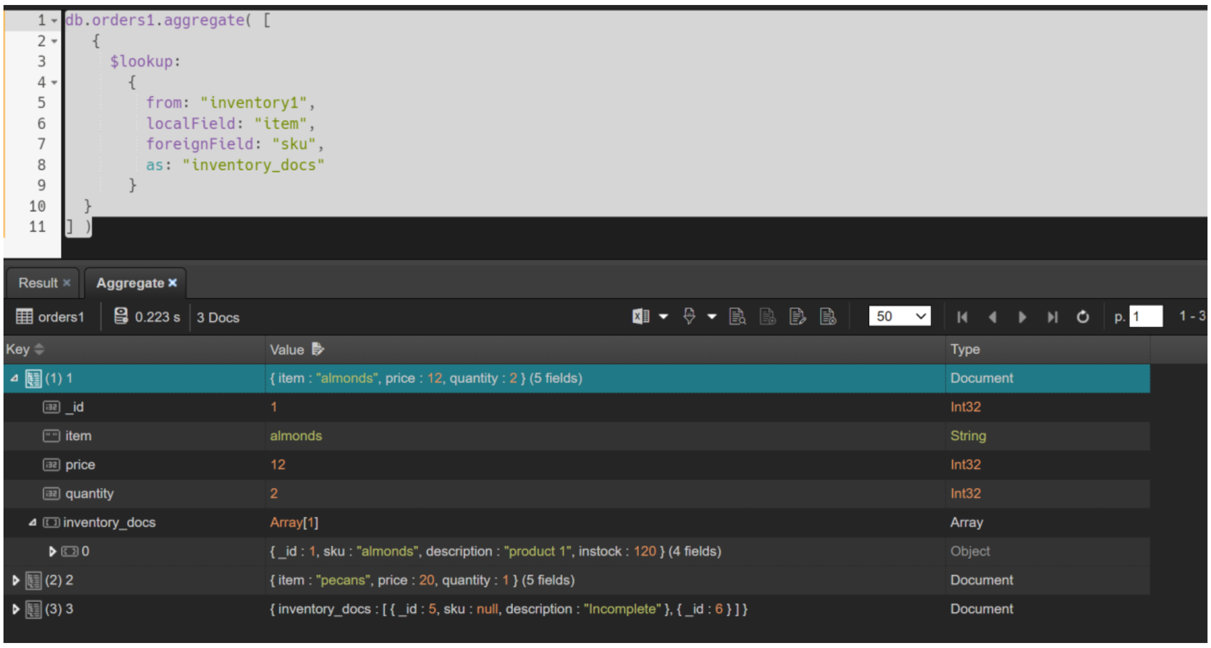
Joins and and $lookup

```
db.orders1.insertMany( [  { "_id" : 1, "item" : "almonds", "price" : 12, "quantity" : 2 },

  { "_id" : 2, "item" : "pecans", "price" : 20, "quantity" : 1 },  { "_id" : 3  }] )
```

```
db.inventory1.insertMany( [  { "_id" : 1, "sku" : "almonds", "description": "product 1", "instock" : 120 }, { "_id" : 2, "sku" : "bread", "description": "product 2", "instock" : 80 },{ "_id" : 3, "sku" : "cashews", "description": "product 3", "instock" : 60 },  { "_id" : 4, "sku" : "pecans", "description": "product 4", "instock" : 70 },{ "_id" : 5, "sku": null, "description": "Incomplete" },  { "_id" : 6 }] )
```

Single equality join with $lookup

```
db.orders1.aggregate( [

  { $lookup:

     {

       from: "inventory1",

       localField: "item",

       foreignField: "sku",

       as: "inventory_docs"

     }

 }] )
```



```
db.orders1.aggregate( [
   {
     $lookup:
       {
         from: "inventory1",
         localField: "item",
         foreignField: "sku",
         as: "inventory_docs"
       }
  }
] )
```

```
Key                    Value                                                                                  Type
(1) 1                  { item : "almonds", price : 12, quantity : 2 } (5 fields)                              Document
    _id                1                                                                                      Int32
    item               almonds                                                                                String
    price              12                                                                                     Int32
    quantity           2                                                                                      Int32
    inventory_docs     Array[1]                                                                               Array
        0              { _id : 1, sku : "almonds", description : "product 1", instock : 120 } (4 fields)      Object
(2) 2                  { item : "pecans", price : 20, quantity : 1 } (5 fields)                               Document
(3) 3                  { inventory_docs : [ { _id : 5, sku : null, description : "Incomplete" }, { _id : 6 } ] } Document
```