IAB206 Assessment 2

SEPTEMBER **15**

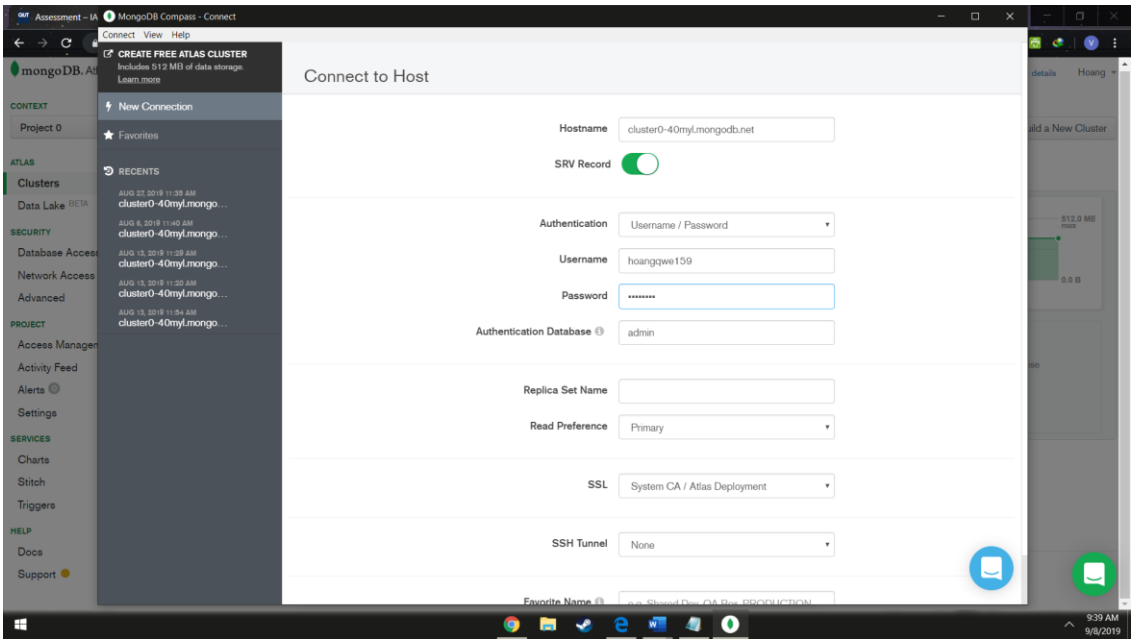
Queensland University of Technology
Authored by: Do Viet Hoang
Nguyen Quoc Huy

Logo Name

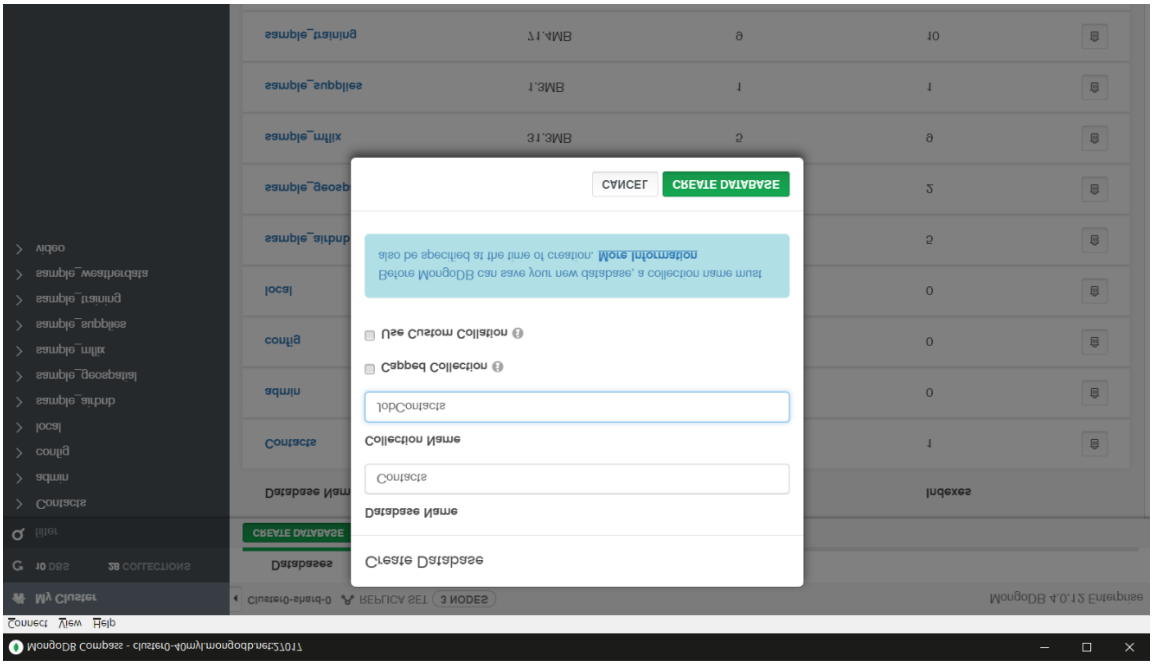
Question 1

CREATE A DATABASE WITH A JUSTIFIABLE NAME IN MONGODB ATLAS CLUSTER.

Connect to Host:



Create Database with a justifiable name



Question 2

THINK OF A DOCUMENT STRUCTURE YOU WOULD LIKE TO HAVE FOR EACH CONTACT.

Name: String
Interest: Array of String
Contact information: Nested Document [Phone: [home, mobile], Email: [Personal, Work]]
Education: Array of String
Languages: Array of String
Meeting Place: Document Type
Organization: Nested Document [Name, Position (array)]
Years of experience: Integer
Skills: Array of String
Overall Impression: String

Question 3

ENTER 20 DOCUMENTS WITH FIELD AND VALUES, BASED ON THE STRUCTURE IDENTIFIED IN QUESTION 2.

```
1   Name : "Evan Smith "                                  String
2  ⌄Interest : Array                                      Array
3     0 : "software development "                         String
4  ⌄Contact Info : Object                                 Object
5    ⌄Phone : Object                                      Object
6        Mobile : "+610003423 "                           String
7        Home : "+610000033 "                             String
8    ⌄Email : Object                                      Object
9        Personal : "evanS@gmail.com "                    String
10       Work : "evanWork@gmail.com "                     String
11 ⌄Education : Array                                     Array
12    0 : "Griffith University "                           String
13    1 : "Brisbane Highschool "                           String
14 ⌄Languages : Array                                     Array
15    0 : "English "                                       String
16    1 : "Spanish "                                       String
17    2 : "Vietnamese "                                    String
18 ⌄Meeting Place : Object                                Object
19    Event : "Career in your hand "                       String
20 ⌄Organization : Object                                 Object
21    Name : "Amazon "                                     String
22    Position : "software developer "                     String
23  Years of experience : 15                               Int32
24 ⌄Skills : Array                                        Array
25    0 : "C# Programming "                                String
26    1 : "process minning "                               String
27    2 : "modern data managementA "                       String
28  Overall Impression : "friendly "                       String
```

Question 4

WRITE A QUERY THAT LOOKS FOR A CONTACT WITH CURRENT ORGANISATION AS SOME UNIVERSITY AND THE POSITION AS LECTURER.

Query:

db.JobContacts.findOne({"Organization.Name":/.University.*/,"Organization.Position": "lecturer"})

Results:

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.JobContacts.findOne({"Organization.Name":/.University.*/,"Organization.P
osition":"lecturer"})
{
        "_id" : ObjectId("5d74553901e48d1dec942c4e"),
        "Name" : "Thomas Lon",
        "Interest" : [
                "coding",
                "teaching"
        ],
        "Contact Info" : {
                "Phone" : {
                        "Mobile" : "+2812781",
                        "Home" : "+748264282"
                },
                "Email" : {
                        "Personal" : "lonl@gmail.com",
                        "Work" : "lonk@gmail.com"
                }
        },
        "Education" : [
                "QUT",
                "Queensland Academy for Science Mathematics and Technology"
        ],
        "Languages" : [
                "English",
                "Vietnamese"
        ],
        "Meeting Place" : {
                "Location" : "QUT",
                "Event" : "how to be good at math",
                "Date" : ISODate("2019-03-11T14:00:00Z"),
                "Time" : "10pm"
        },
        "Organization" : [
                {
                        "Name" : "Ebay",
                        "Position" : "hardware manager"
                },
                {
                        "Name" : "Grifith University",
                        "Position" : "lecturer"
                }
        ],
        "Years of experience" : 12,
        "Skills" : [
                "accounting",
                "data mining"
        ],
        "Overall Impression" : "normal"
```

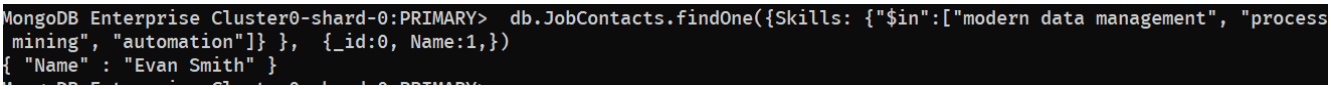
Question 5

WRITE A QUERY EXTRACTING THE NAME OF AN INDIVIDUAL WHO HAS AT LEAST ONE OF THESE SKILLS: ‘MODERN DATA MANAGEMENT’, ‘PROCESS MINING’, AND ‘AUTOMATION’.

Query:

db.JobContacts.findOne({Skills: {"$in":["modern data management", "process mining", "automation"]} }, {_id:0, Name:1,})

Result:



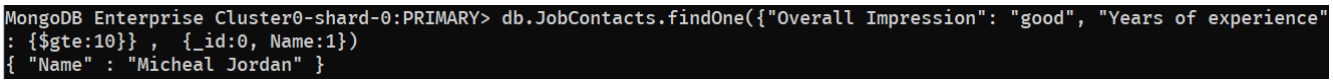
Question 6

WRITE A QUERY EXTRACTING THE NAME OF AN INDIVIDUAL WHOSE OVERALL IMPRESSION IS ‘GOOD’ AND YEARS OF EXPERIENCE IS GREATER THAN OR EQUAL TO 10.

Query:

db.JobContacts.findOne({"Overall Impression": "good", "Years of experience": {$gte:10}} , {_id:0, Name:1})

Result:



Question 7

FIND NAME AND CONTACT INFO OF PERSONS WHO STUDIED AT “QUT” AND HAVE SKILL IN “DATA SCIENCE” (DO VIET HOANG)

Query:

db.JobContacts.find({"Education": "QUT", "Skills": "data science"}, {_id:0, Name:1, "Contact Info": 1}).pretty()

Result:

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.JobContacts.find({"Education": "QUT", "Skills": "data science"}, {_id:0,
 Name:1, "Contact Info": 1}).pretty()
{
        "Name" : "Anna Pham",
        "Contact Info" : {
                "Phone" : {
                        "Mobile" : "+88712376",
                        "Home" : "+3114872"
                },
                "Email" : {
                        "Personal" : "annal@gmail.com",
                        "Work" : "annak@gmail.com"
                }
        }
}
```

FIND NAME AND CONTACT INFO OF PERSONS WHO HAVE MORE THAN 5 YEARS OF EXPERIENCE IN ARTIFICIAL INTELLIGENCE (NGUYEN QUOC HUY)

Query:

db.JobContacts.find({"Skills":"artificial intelligence", "Years of experience":{$gt:5}}, {_id:0, Name:1, "Contact Info": 1}).pretty()

Result:

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.JobContacts.find({"Skills":"artificial intelligence", "Years of experien
ce":{$gt:5}}, {_id:0, Name:1, "Contact Info": 1}).pretty()
{
        "Name" : "David John",
        "Contact Info" : {
                "Phone" : {
                        "Mobile" : "+12473638",
                        "Home" : "+92874728"
                },
                "Email" : {
                        "Personal" : "johnl@gmail.com",
                        "Work" : "johnk@gmail.com"
                }
        }
}
```

Question 8

WRITE A QUERY TO UPDATE THE EMAIL ADDRESS OF A CONTACT THAT YOU HAVE IN YOUR DATABASE

Query:

db.JobContacts.updateOne({"Name": "Harvey Specter"},{$set: {"Contact Info.Email.Personal": "harveyspecter123@gmail.com"}})

Result:

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.JobContacts.updateOne({"Name": "Harvey Specter"},{$set: {"Contact Info.E
mail.Personal": "harveyspecter123@gmail.com"}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

Before


```
_id: ObjectId("5d74511fc3031848a8728080")
Name: "Harvey Specter"
Interest: Array
    0: "poker"
    1: "watching movies"
Contact Info: Object
  Phone: Object
      Mobile: "+6172345677"
      Home: "+612345265"
  Email: Object
      Personal: "harvey@gmail.com"
      Work: "harveyWork@gmail.com"
Education: Array
    0: "Havard Law University"
Languages: Array
    0: "English"
    1: "France"
Meeting Place: Object
    Location: "QUT"
Organization: Object
    Name: "Pearson Specter Litt"
    Position: "data administrator"
Years of experience: 30
Skills: Array
    0: "databases"
    1: "deep learning"
Overall Impression: "good"
```


After


```
_id: ObjectId("5d74511fc3031848a8728080")
Name: "Harvey Specter"
Interest: Array
    0: "poker"
    1: "watching movies"
Contact Info: Object
  Phone: Object
      Mobile: "+6172345677"
      Home: "+612345265"
  Email: Object
      Personal: "harveyspecter123@gmail.com"
      Work: "harveyWork@gmail.com"
Education: Array
    0: "Havard Law University"
Languages: Array
    0: "English"
    1: "France"
Meeting Place: Object
    Location: "QUT"
Organization: Object
    Name: "Pearson Specter Litt"
    Position: "data administrator"
Years of experience: 30
Skills: Array
    0: "databases"
    1: "deep learning"
Overall Impression: "good"
```

Question 9

WRITE A QUERY TO INCREMENT THE YEARS OF EXPERIENCE OF A CONTACT WITH CURRENT ORGANISATION OF YOUR CHOICE BY 2.

Query:

db.JobContacts.updateOne({"Organization.Name": "Amazon"}, {$inc: {"Years of experience": 2}})

Result:

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.JobContacts.updateOne({"Organization.Name": "Amazon"}, {$inc: {"Years of
 experience": 2}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
MongoDB Enterprise Cluster0-shard-0:PRIMARY>
```

Before


```
_id: ObjectId("5d744961c3031848a872807a")
Name: "Evan Smith"
Interest: Array
    0: "software development"
Contact Info: Object
  Phone: Object
      Mobile: "+610003423"
      Home: "+610000033"
  Email: Object
      Personal: "evanS@gmail.com"
      Work: "evanWork@gmail.com"
Education: Array
    0: "Griffith University"
    1: "Brisbane Highschool"
Languages: Array
    0: "English"
    1: "Spanish"
    2: "Vietnamese"
Meeting Place: Object
    Event: "Career in your hand"
Organization: Object
    Name: "Amazon"
    Position: "software developer"
Years of experience: 15
Skills: Array
    0: "C# Programming"
    1: "process minning"
    2: "modern data management"
Overall Impression: "friendly"
```


After


```
_id: ObjectId("5d744961c3031848a872807a")
Name: "Evan Smith"
Interest: Array
    0: "software development"
Contact Info: Object
  Phone: Object
      Mobile: "+610003423"
      Home: "+610000033"
  Email: Object
      Personal: "evanS@gmail.com"
      Work: "evanWork@gmail.com"
Education: Array
    0: "Griffith University"
    1: "Brisbane Highschool"
Languages: Array
    0: "English"
    1: "Spanish"
    2: "Vietnamese"
Meeting Place: Object
    Event: "Career in your hand"
Organization: Object
    Name: "Amazon"
    Position: "software developer"
Years of experience: 17
Skills: Array
    0: "C# Programming"
    1: "process minning"
    2: "modern data management"
Overall Impression: "friendly"
```

Question 10

WRITE A QUERY TO UPDATE THE MEETING PLACE, DATE, AND TIME OF A CONTACT OF YOUR CHOICE.

Query:

db.JobContacts.updateOne({Name: "Edward Ben"}, {$set:{"Meeting Place.Location": "Modern Data Management class", "Meeting Place.Date": new Date("2016-05-18T16:30:00Z")}})

Result:

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.JobContacts.updateOne({Name: "Edward Ben"}, {$set:{"Meeting Place.Locati
on": "Modern Data Management class", "Meeting Place.Date": new Date("2016-05-18T16:30:00Z")}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

Before


```
_id: ObjectId("5d7450d501e48d1dec942c4a")
Name: "Edward Ben"
Interest: Array
    0: "football"
    1: "beatbox"
Contact Info: Object
  Phone: Object
      Mobile: "+273622"
      Home: "+988764467"
  Email: Object
      Personal: "benl@gmail.com"
      Work: "benw@gmail.com"
Education: Array
    0: "Queensland University"
    1: "Marist College Ashgrove"
Languages: Array
    0: "English"
    1: "Portuguese"
Meeting Place: Object
    Event: "PHP consultation"
    Location: "QUT"
Organization: Object
    Name: "Kiwi"
    Position: "web developer"
Years of experience: 15
Skills: Array
    0: "front-end developing"
    1: "automation"
Overall Impression: "good"
```


After


```
_id: ObjectId("5d7450d501e48d1dec942c4a")
Name: "Edward Ben"
Interest: Array
    0: "football"
    1: "beatbox"
Contact Info: Object
  Phone: Object
      Mobile: "+273622"
      Home: "+988764467"
  Email: Object
      Personal: "benl@gmail.com"
      Work: "benw@gmail.com"
Education: Array
    0: "Queensland University"
    1: "Marist College Ashgrove"
Languages: Array
    0: "English"
    1: "Portuguese"
Meeting Place: Object
    Event: "PHP consultation"
    Location: "Modern Data Management class"
    Date: 2016-05-18T16:30:00.000+00:00
Organization: Object
    Name: "Kiwi"
    Position: "web developer"
Years of experience: 15
Skills: Array
    0: "front-end developing"
    1: "automation"
Overall Impression: "good"
```

Question 11

UPDATE THE VALUE OF SKILL OF A CONTACT OF YOUR CHOICE FROM ‘DATABASES’ TO ‘MODERN DATABASES’.

Query:

db.JobContacts.updateOne({Name: "David John", Skills: "databases"}, {$set:{"Skills.$": "modern databases"}})

Result:

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.JobContacts.updateOne({Name: "David John", Skills: "databases"}, {$set:{
"Skills.$": "modern databases"}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

Before

```
_id: ObjectId("5d744fb901e48d1dec942c49")
Name: "David John"
Interest: Array
    0: "jogging"
    1: "swimming"
Contact Info: Object
  Phone: Object
      Mobile: "+12473638"
      Home: "+92874728"
  Email: Object
      Personal: "johnl@gmail.com"
      Work: "johnk@gmail.com"
Education: Array
    0: "Kelvin Grove State College"
    1: "Griffith University"
Languages: Array
    0: "English"
    1: "Japanese"
Meeting Place: Object
    Event: "job consultation"
    Location: "QUT"
Organization: Object
    Name: "Netflix"
    Position: "application developer"
Years of experience: 18
Skills: Array
    0: "databases"
    1: "artificial intelligence"
Overall Impression: "bad"
```

After

```
_id: ObjectId("5d744fb901e48d1dec942c49")
Name: "David John"
Interest: Array
    0: "jogging"
    1: "swimming"
Contact Info: Object
  Phone: Object
      Mobile: "+12473638"
      Home: "+92874728"
  Email: Object
      Personal: "johnl@gmail.com"
      Work: "johnk@gmail.com"
Education: Array
    0: "Kelvin Grove State College"
    1: "Griffith University"
Languages: Array
    0: "English"
    1: "Japanese"
Meeting Place: Object
    Event: "job consultation"
    Location: "QUT"
Organization: Object
    Name: "Netflix"
    Position: "application developer"
Years of experience: 18
Skills: Array
    0: "modern databases"
    1: "artificial intelligence"
Overall Impression: "bad"
```

Question 12

FIND A CONTACT WITH THE CURRENT ORGANIZATION AS ‘QUT’ AND UPDATE THE SKILLS TO ADD A NEW SKILL: ‘AUTOMATION’.

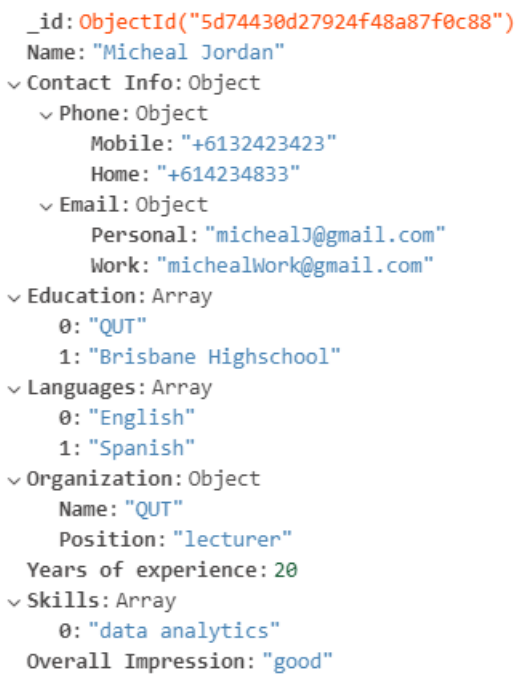
Query:

db.JobContacts.updateOne({"Organization.Name": "QUT"}, {$addToSet:{Skills: "automation"}})

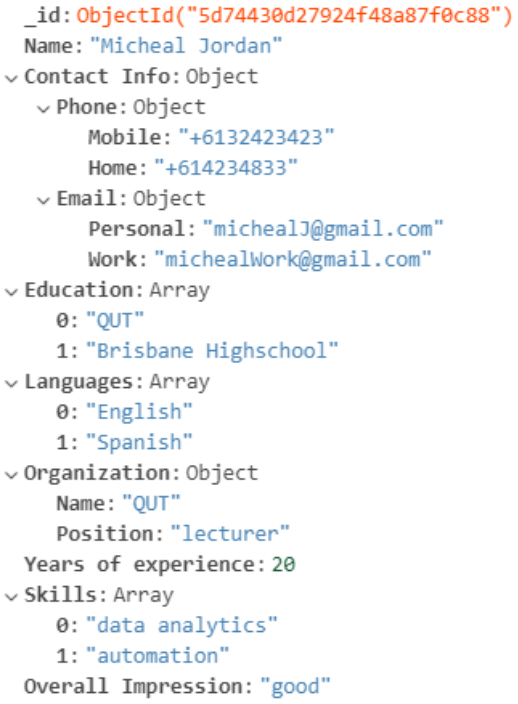
Result:

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.JobContacts.updateOne({"Organization.Name": "QUT"}, {$addToSet:{Skills:
"automation"}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

Before



After



Question 13

UPDATE OVERALL IMPRESSION OF PERSONS WHO WORK IN ORGANIZATION "QUT" TO "GOOD".

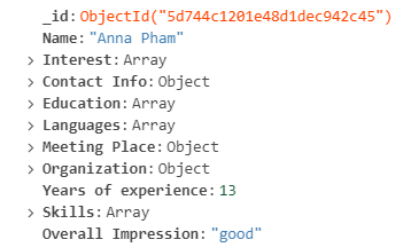
Query:

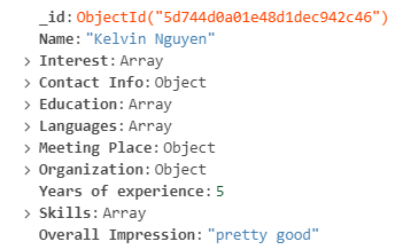
db.JobContacts.updateMany({"Organization.Name": "QUT"},{$set:{"Overall Impression":"good"}})

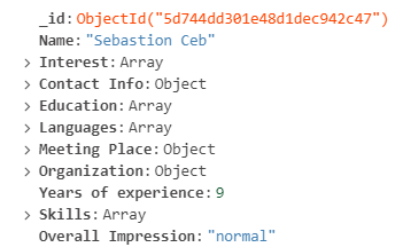
Result:

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.JobContacts.updateMany({"Organization.Name": "QUT"},{$set:{"Overall Impr
ession":"good"}})
{ "acknowledged" : true, "matchedCount" : 4, "modifiedCount" : 2 }
```

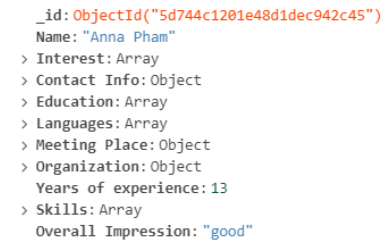
Before

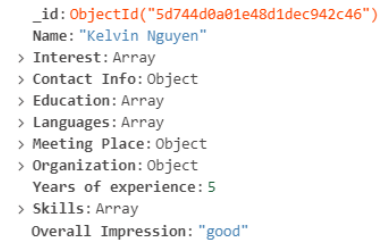


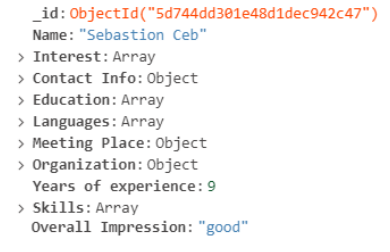




After







Question 14

DELETE A DOCUMENT (I.E. DETAILS OF A CONTACT) OF YOUR CHOICE USING A SPECIFIC CONDITION.

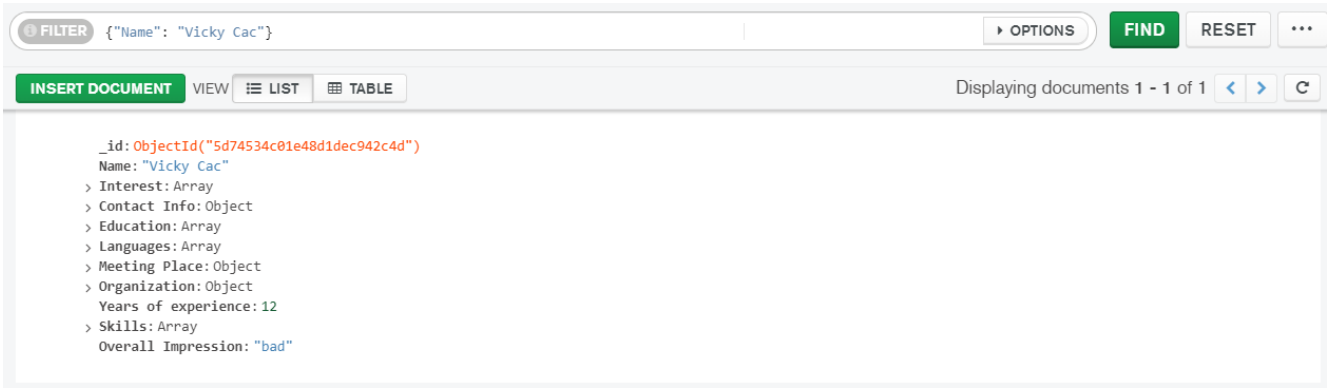
Query:

db.JobContacts.deleteOne({"Name": "Vicky Cac"})

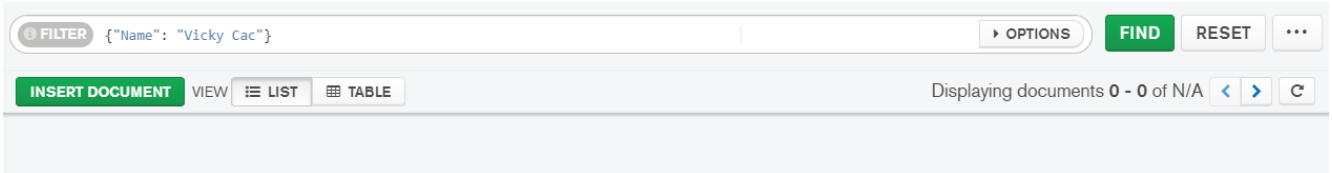
Result:

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.JobContacts.deleteOne({"Name": "Vicky Cac"})
{ "acknowledged" : true, "deletedCount" : 1 }
```

Before:



After:



Question 15

DELETE ALL DOCUMENTS RELATING TO AN ORGANIZATION OF YOUR CHOICE.

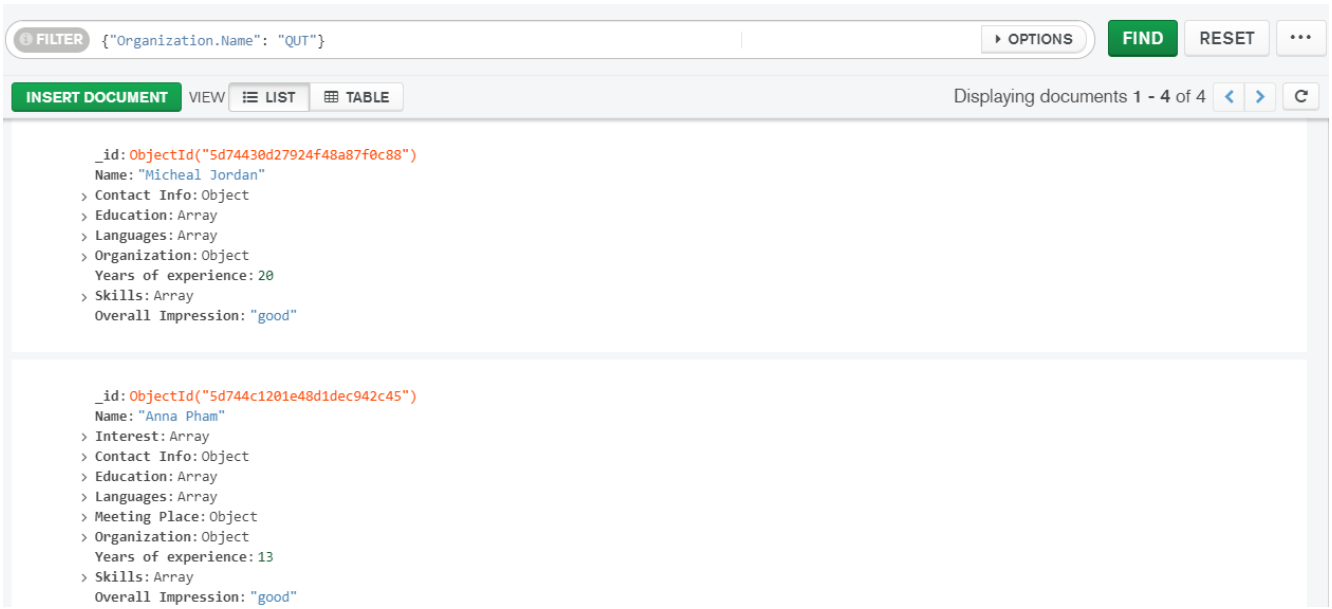
Query:

db.JobContacts.remove({"Organization.Name": "QUT"})

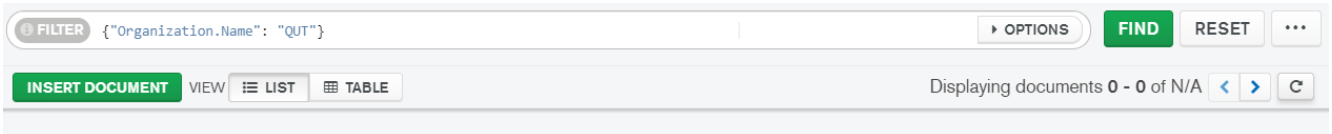
Result:

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.JobContacts.remove({"Organization.Name": "QUT"})
WriteResult({ "nRemoved" : 4 })
```

Before:



After:



Question 16

SCENARIO

Blablabla

DOCUMENT STRUCTURE

Title: String
Author: String
Genre: Array of String
Publisher: Nested Document [Name (string), Year (integer), Country (array)]
Format: Array of String
Quantity: Integer
Price: Decimal

THREE CREATE QUERIES

Query 1

Purpose: Insert a new book

Query:

db.Books.insertOne({"title":"The Secret of Naoko","author": "Suga Keigo", genre :["detective", "drama", "mystery"],"publisher":{ name: "Kadogawa Book", year: 2006, country : "Japan"},"format":["hardcover", "ebook"], quantity: 100, price: 19.99 })

Result:

```
MongoDB Enterprise > db.Books.insertOne({"title":"The Secret of Naoko","author": "Suga Keigo", genre :["detective", "dra
ma", "mystery"],"publisher":{ name: "Kadogawa Book", year: 2006, country : "Japan"},"format":["hardcover", "ebook"], qua
ntity: 100, price: 19.99 })
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5d7878d50982df6e6c6c9ff6")
}
```


```
_id: ObjectId("5d7878d50982df6e6c6c9ff6")
title: "The Secret of Naoko"
author: "Suga Keigo"
genre: Array
    0: "detective"
    1: "drama"
    2: "mystery"
publisher: Object
    name: "Kadogawa Book"
    year: 2006
    country: "Japan"
format: Array
    0: "hardcover"
    1: "ebook"
quantity: 100
price: 19.99
```

Query 2

Purpose: Insert two new books

Query:

db.Books.insertMany([{"title":"Sleeping Princess","author": "Sebastian Ceb", genre :["romance", "history"],"publisher":{ name: "Peter Harison", year: 2002, country : "United States"},"format":["hardcover", "paperback"], quantity: 20, price: 34.99 }, {"title":"Little Man","author": "Danny Thomath", genre :["novel"],"publisher":{ name: "Hackerman Media", year: 1990, country : "Germany"},"format":["hardcover", "ebook"], quantity: 28, price: 45.99 }])

Result:

```
MongoDB Enterprise > db.Books.insertMany([ {"title":"Sleeping Princess","author": "Sebastian Ceb", genre :["romance", "h
istory"],"publisher":{ name: "Peter Harison", year: 2002, country : "United States"},"format":["hardcover", "paperback"]
, quantity: 20, price: 34.99 }, {"title":"Little Man","author": "Danny Thomath", genre :["novel"],"publisher":{ name: "H
ackerman Media", year: 1990, country : "Germany"},"format":["hardcover", "ebook"], quantity: 28, price: 45.99 } ])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5d787aaa0982df6e6c6c9ff7"),
                ObjectId("5d787aaa0982df6e6c6c9ff8")
        ]
}
```

```
_id: ObjectId("5d787aaa0982df6e6c6c9ff7")
title: "Sleeping Princess"
author: "Sebastian Ceb"
genre: Array
    0: "romance"
    1: "history"
publisher: Object
    name: "Peter Harison"
    year: 2002
    country: "United States"
format: Array
    0: "hardcover"
    1: "paperback"
quantity: 20
price: 34.99
```

```
_id: ObjectId("5d787aaa0982df6e6c6c9ff8")
title: "Little Man"
author: "Danny Thomath"
genre: Array
    0: "novel"
publisher: Object
    name: "Hackerman Media"
    year: 1990
    country: "Germany"
format: Array
    0: "hardcover"
    1: "ebook"
quantity: 28
price: 45.99
```

Query 3

Purpose: Insert a Vietnamese book

Query:

db.Books.insertOne({"title":"Tam Cam","author": "Vietnamese Community", genre :["fiction", "folk-tale"],"publisher":{ name: "Kim Dong", year: 1901, country : "Viet Nam"},"format":["hardcover", "paperback"], quantity: 10, price: 4.99})

Result:

```
MongoDB Enterprise > db.Books.insertOne({"title":"Tam Cam","author": "Vietnamese Community", genre :["fiction", "folk-ta
le"],"publisher":{ name: "Kim Dong", year: 1901, country : "Viet Nam"},"format":["hardcover", "paperback"], quantity: 10
, price: 4.99})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5d787cb10982df6e6c6c9ff9")
}
```

```
_id: ObjectId("5d787cb10982df6e6c6c9ff9")
title: "Tam Cam"
author: "Vietnamese Community"
genre: Array
    0: "fiction"
    1: "folk-tale"
publisher: Object
    name: "Kim Dong"
    year: 1901
    country: "Viet Nam"
format: Array
    0: "hardcover"
    1: "paperback"
quantity: 10
price: 4.99
```

THREE READ QUERIES

Query 1

Purpose: Look for title and author of the books from Japan and published year after 1990

Query:

db.Books.find({"publisher.country": "Japan", "publisher.year" : {$gt: 1900}}, {_id: 0, title: 1, author: 1}).pretty()

Result:

```
MongoDB Enterprise > db.Books.find( {"publisher.country": "Japan", "publisher.year" : {$gt: 1900}}, {_id: 0, title: 1, a
uthor: 1}).pretty()
{ "title" : "Starting Over", "author" : "Sugaru Miaki" }
{ "title" : "Pain Pain Go Away", "author" : "Sugaru Miaki" }
{ "title" : "The Secret of Naoko", "author" : "Suga Keigo" }
```

Query 2

Purpose: Look for books has format paperback and quantity is 0

Query:

db.Books.find({format: "paperback", quantity: 0 }, {_id: 0}).pretty()

Result:

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.Books.find({format: "paperback", quantity: 0 }, {_i
d: 0}).pretty()
{
        "title" : "Pain Pain Go Away",
        "author" : "Sugaru Miaki",
        "genre" : [
                "drama",
                "horror",
                "tragedy"
        ],
        "publisher" : {
                "name" : "Ascii Media Works",
                "year" : 2014,
                "country" : "Japan"
        },
        "format" : [
                "paperback",
                "ebook"
        ],
        "quantity" : 0,
        "price" : 25.99
}
```

Query 3

Purpose: Look for title, author and genres of the books with genre is "romance" or "detective"

Query:

db.Books.find({genre: {"$in":["romance", "detective"]} }, {_id: 0, title: 1, author: 1, genre:1}).pretty()

Result:

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.Books.find({genre: {"$in":["romance", "detective"]}
 }, {_id: 0, title: 1, author: 1, genre:1}).pretty()
{
        "title" : "Starting Over",
        "author" : "Sugaru Miaki",
        "genre" : [
                "drama",
                "mature",
                "romance",
                "slice of life"
        ]
}
{
        "title" : "Brooklyn Girl",
        "author" : "Guillaume Musso",
        "genre" : [
                "novel",
                "detective"
        ]
}
{
        "title" : "Me Before You",
        "author" : "Jojo Moyes",
        "genre" : [
                "romance",
                "fiction"
        ]
}
```

```
{
        "title" : "The Secret of Naoko",
        "author" : "Suga Keigo",
        "genre" : [
                "detective",
                "drama",
                "mystery"
        ]
}
{
        "title" : "Sleeping Princess",
        "author" : "Sebastian Ceb",
        "genre" : [
                "romance",
                "history"
        ]
}
```

THREE UPDATE QUERIES

Query 1

Purpose: Increase the quantity of the book named "The Great Gasby" by 10

Query:

db.Books.updateOne({title: "The Great Gasby"}, {$inc : {quantity : 10}})

Result:

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.Books.updateOne({title: "The Great Gasby"}, {$inc :
 {quantity : 10}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

Before


```
_id: ObjectId("5d78720e6d3f7d3730db2579")
title: "The Great Gasby"
author: "Scott Fitzgeraldi"
genre: Array
    0: "novel"
publisher: Object
    name: "Charles Scribner's Sons"
    year: 1925
    country: "United States"
format: Array
    0: "paperback"
    1: "ebook"
    2: "hardcover "
quantity: 100
price: 60.99
```


After


```
_id: ObjectId("5d78720e6d3f7d3730db2579")
title: "The Great Gasby"
author: "Scott Fitzgeraldi"
genre: Array
    0: "novel"
publisher: Object
    name: "Charles Scribner's Sons"
    year: 1925
    country: "United States"
format: Array
    0: "paperback"
    1: "ebook"
    2: "hardcover "
quantity: 110
price: 60.99
```

Query 2

Purpose: Add the format "Ebook" and "paperback" to all the books that has the genre "romance"

Query:

db.Books.updateMany({genre: "romance"}, {$addToSet:{format:{$each: ["ebook", "paperback"]}}})

Result:

```
MongoDB Enterprise > db.Books.updateMany({genre: "romance"}, {$addToSet:{format:{$each: ["ebook", "
paperback"]}}})
{ "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 1 }
```


```
_id: ObjectId("5d786c16bc3b113730e1cd8e")
title: "Starting Over"
author: "Sugaru Miaki"
> genre: Array
> publisher: Object
v format: Array
    0: "paperback"
    1: "ebook"
quantity: 30
price: 19.99
```

```
_id: ObjectId("5d786c16bc3b113730e1cd8e")
title: "Starting Over"
author: "Sugaru Miaki"
> genre: Array
> publisher: Object
v format: Array
    0: "paperback"
    1: "ebook"
quantity: 30
price: 19.99
```

```
_id: ObjectId("5d7872a76d3f7d3730db257b")
title: "Me Before You"
author: "Jojo Moyes"
> genre: Array
> publisher: Object
v format: Array
    0: "paperback"
    1: "ebook"
    2: "hardcover "
quantity: 80
price: 130.99
```

```
_id: ObjectId("5d7872a76d3f7d3730db257b")
title: "Me Before You"
author: "Jojo Moyes"
> genre: Array
> publisher: Object
v format: Array
    0: "paperback"
    1: "ebook"
    2: "hardcover "
quantity: 80
price: 130.99

_id: ObjectId("5d787aaa0982df6e6c6c9ff7")
title: "Sleeping Princess"
author: "Sebastian Ceb"
v genre: Array
    0: "romance"
    1: "history"
v publisher: Object
    name: "Peter Harison"
    year: 2002
    country: "United States"
v format: Array
    0: "hardcover"
    1: "paperback"
    2: "ebook"
quantity: 20
price: 34.99
```

```
_id: ObjectId("5d787aaa0982df6e6c6c9ff7")
title: "Sleeping Princess"
author: "Sebastian Ceb"
> genre: Array
> publisher: Object
v format: Array
    0: "hardcover"
    1: "paperback"
quantity: 20
price: 34.99
```

Query 3

Purpose: Decrease the price by 25% for every book from "United States"

Query:

db.Books.updateMany({"publisher.country": "United States"}, {$mul: {"price": NumberDecimal("0.75")}})

Result:

```
MongoDB Enterprise > db.Books.updateMany({"publisher.country": "United States"}, {$mul: {"price": N
umberDecimal("0.75")}})
{ "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 3 }
```


```
_id: ObjectId("5d78720e6d3f7d3730db2579")
title: "The Great Gasby"
author: "Scott Fitzgeraldi"
genre: Array
publisher: Object
    name: "Charles Scribner's Sons"
    year: 1925
    country: "United States"
format: Array
quantity: 110
price: 60.99
```

```
_id: ObjectId("5d78720e6d3f7d3730db2579")
title: "The Great Gasby"
author: "Scott Fitzgeraldi"
genre: Array
publisher: Object
    name: "Charles Scribner's Sons"
    year: 1925
    country: "United States"
format: Array
quantity: 110
price: 45.742500000000000
```

```
_id: ObjectId("5d7872f06d3f7d3730db257c")
title: "How to learn Modern Data Management"
author: "Louis Marllow Litt"
genre: Array
publisher: Object
    name: "Person Specter"
    year: 2003
    country: "United States"
format: Array
quantity: 50
price: 70.99
```

```
_id: ObjectId("5d7872f06d3f7d3730db257c")
title: "How to learn Modern Data Management"
author: "Louis Marllow Litt"
genre: Array
publisher: Object
    name: "Person Specter"
    year: 2003
    country: "United States"
format: Array
quantity: 50
price: 53.242500000000000
```

```
_id: ObjectId("5d787aaa0982df6e6c6c9ff7")
title: "Sleeping Princess"
author: "Sebastian Ceb"
genre: Array
publisher: Object
    name: "Peter Harison"
    year: 2002
    country: "United States"
format: Array
quantity: 20
price: 34.99
```

```
_id: ObjectId("5d787aaa0982df6e6c6c9ff7")
title: "Sleeping Princess"
author: "Sebastian Ceb"
genre: Array
    0: "romance"
    1: "history"
publisher: Object
    name: "Peter Harison"
    year: 2002
    country: "United States"
format: Array
    0: "hardcover"
    1: "paperback"
    2: "ebook"
quantity: 20
price: 26.242500000000000
```

ONE DELETE QUERY

Query 1

Purpose: Remove all the books that are too old (published before 2000)

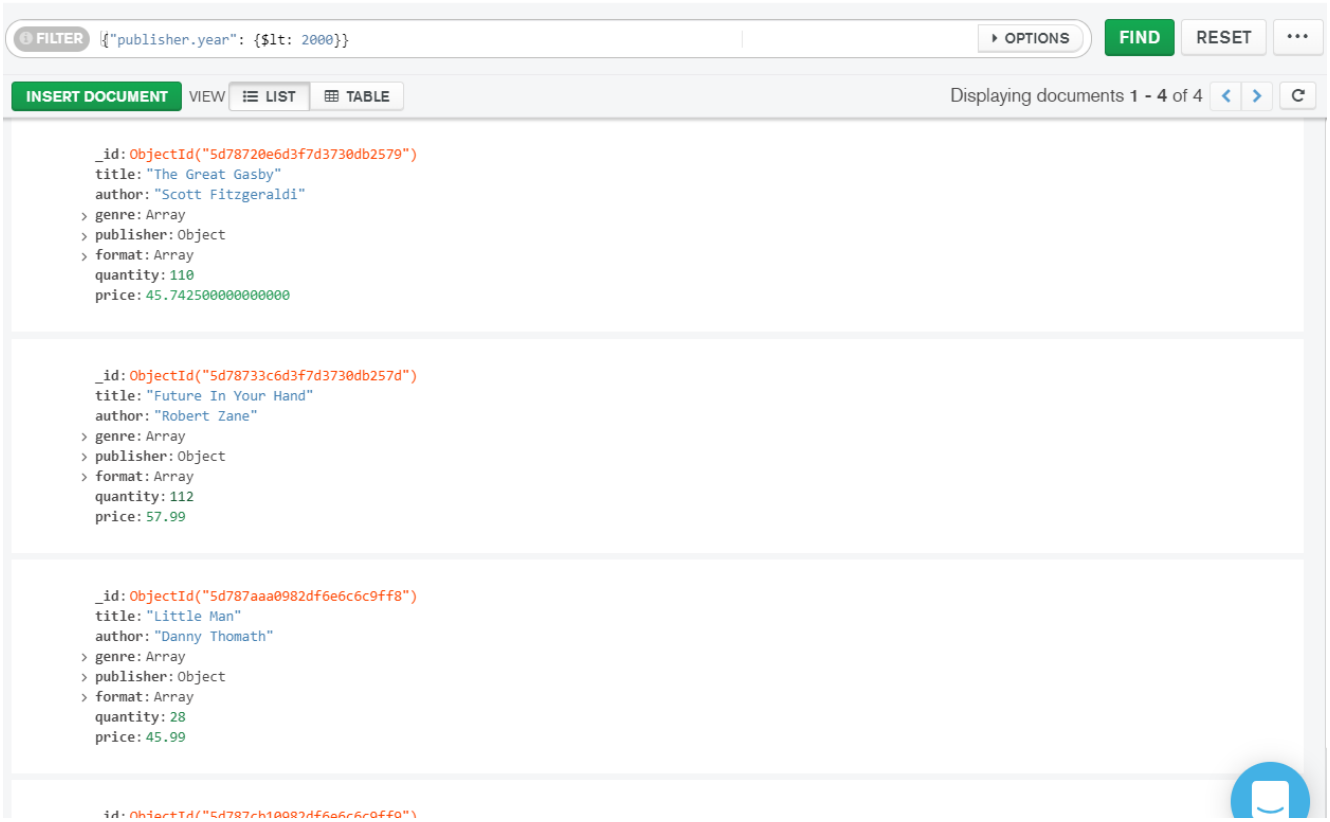
Query:

db.Books.remove({"publisher.year": {$lt: 2000}})

Result:

```
MongoDB Enterprise > db.Books.remove({"publisher.year": {$lt: 2000}})
WriteResult({ "nRemoved" : 4 })
```

Before



After: