

**Declaration:** I have viewed the final version of the assignment that is to be submitted and it is my original work.

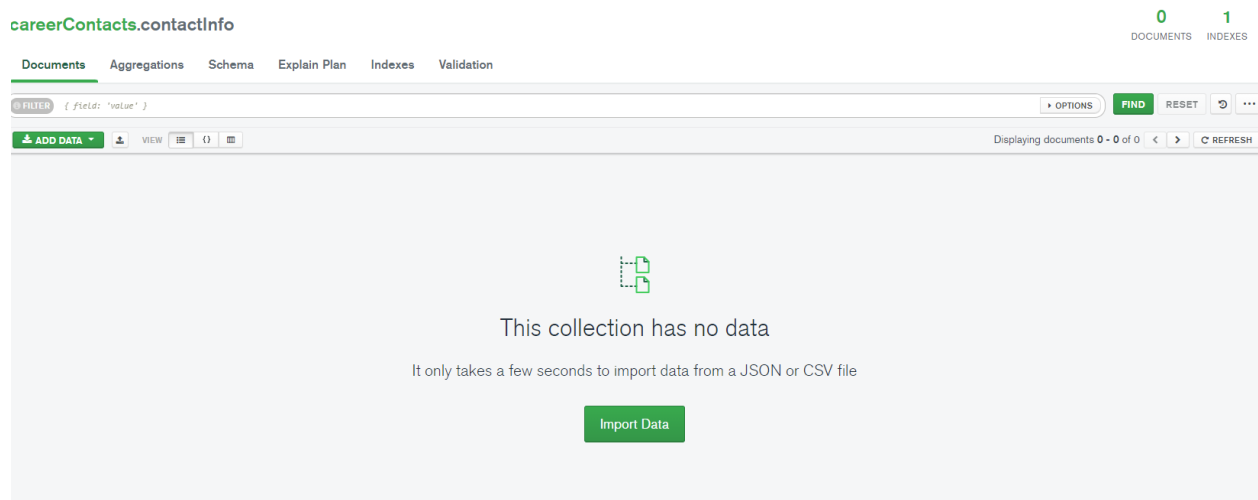
**Signature:** Irfan Rashad, Nicholas Faleao, Andrew Anderson, Minjae Lee

# Project 1

## Part A:

**Create:**

**Database:**



**Document Structure:**

Field	Description	Type
name	Name of the contact.	String
meeting_place	The place where you met the contact and other details (event, date, time etc).	Document Type[location (string), date (string), event (string)]
age	Age of contact	Integer

organisation	All the organisations contact has worked in. The first document in the array is the contact's current organisation.	Array of documents [{name(string), position(string),...}]
skills	The skills possessed by the contact.	Array of String
education	Education institutions the contact has attended.	Array of String
qualifications	The contact's certified qualifications (e.g. degrees).	Array of String
experience	The years of experience the contact has in the industry.	Integer
projects	List of work done by the contacts	Array of string
accolades	Number of achievements	Integer
contact_number	Phone number	String
contact_email	Work and personal email address	Document type[work, personal]
related_contacts	Contacts related to the original contact	Array of string
impression	Your overall impression of the contact in terms of his likelihood of helping you and the area he is working in.	String

## Documents:

[careerContacts.contactInfo](#)

0  
DOCUMENTS

1  
INDEXES

[Documents](#) [Aggregations](#) [Schema](#) [Explain Plan](#) [Indexes](#) [Validation](#)

**FILTER** { field: 'value' }

[▶ OPTIONS](#)
[FIND](#)
[RESET](#)
[↺](#)
[...](#)

 **ADD DATA**
 **VIEW**




Displaying documents 0 - 0 of 0 [<](#) [>](#) [REFRESH](#)



This collection has no data

It only takes a few seconds to import data from a JSON or CSV file

## Import Data

[illegible]

```

_id: ObjectId('631d39dald896dc2e4410431')
name: "Ravihansa Rajapakse"
meeting_place: Object
  location: "QUT Gardens Point Campus"
  date: "04/08/2022"
  age: 32
organisation: Array
  0: Object
    name: "Higher Education Academy"
    position: "Associate Fellow"
skills: Array
  0: "Modern Data Management"
education: Array
  0: "QUT"
  1: "Perform Australia"
  2: "University of Columbo"
qualifications: Array
  0: "Doctor of Philosophy (PhD) Computer Science (Human Computer Interaction..."
  1: "Certificate IV in Acting for Stage & Screen "
  2: "Bachelor's degree: Information and Communication Technology"
experience: 19
projects: Array
  0: "n/a"
  accolades: 5
  contact_number: "0413014820"
contact_email: Object
  work: "saumya.rajapakse@qut.edu.au"
related_contacts: Array
  0: "Kanika Goel"
  impression: "good"

```

```
_id: ObjectId('631d39da1d896dc2e4410438')
name: "Emmeline Preece"
meeting_place: Object
  location: "QUT Gardens Point Campus"
  date: "20/07/2021"
  event: "Orientation"
age: 33
organisation: Array
  0: Object
    name: "Queensland University of Technology"
    position: "Lecturer"
  1: Object
    name: "AIA Australia Limited"
    position: "Data Analyst"
skills: Array
  0: "Data Analytics"
  1: "Modern Data Management"
education: Array
  0: "Queensland University of Technology"
qualifications: Array
  0: "PhD in Data Science"
  1: "Bachelor of Computer Science"
experience: 11
projects: Array
  0: "Voice Recognition"
  1: "Sound Recognition"
accolades: 2
contact_number: "0491570156"
contact_email: Object
  personal: "emmelinepreece@gmail.com"
  work: "e.preece@qut.edu.au"
related_contacts: Array
  0: "Zayne Bellamy"
  1: "Safiya Marsh"
impression: "excellent"
```

```

    _id: ObjectId('631d3c381d896dc2e4410444')
    name: "Bradley Cooper"
    ✓ meeting_place: Object
      location: "Zoom"
      date: "31/04/2020"
      event: "Project Collaboration"
    age: 24
    ✓ organisation: Array
      ✓ 0: Object
        name: "Figma"
        position: "Full Stack Developer"
      ✓ 1: Object
        name: "JetBrains"
        position: "Front End Developer"
    ✓ skills: Array
      0: "Programming"
      1: "Web Development"
      2: "Graphic Design"
      3: "Databases"
    ✓ education: Array
      0: "Queensland University of Technology"
    ✓ qualifications: Array
      0: "Bachelor of Information Systems"
    experience: 2
    ✓ projects: Array
      0: "Personal content management system"
      1: "Geospatial data logging"
    accolades: 0
    contact_number: "0431224236"
    ✓ contact_email: Object
      personal: "brad.cooper@gmail.com"
      work: "bradleycooper@figma.com"
    ✓ related_contacts: Array
      0: "Tabitha Rodgers"
    impression: "excellent"

```

Except for 2 documents with missing fields, the rest follow the same document structure with all the fields filled in.

## Read:

### Question 4.

```

db.contactInfo.findOne({$and:[{"organisation.0.name":
/University/},{ "organisation.0.position": "Lecturer"}, {"skills": "Data Analytics"}]})

```

Find the first document with the current organisation as some University, the position as lecturer, and with 'data analytics' as skill.

## careerContacts.contactInfo

30 1  
DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter: `{ $and: [{ "organisation.0.name": "/University/" }, { "organisation.0.position": "Lecturer" }, { "skills": "Data Analytics" } ] }` OPTIONS FIND RESET ↻ ⋮

ADD DATA VIEW 0

Displaying documents 1 - 2 of 2 < > C REFRESH

```
{
  "_id": ObjectId("631d39da1d896dc2e4410438"),
  "name": "Emmeline Preece",
  "meeting_place": Object,
  "age": 33,
  "organisation": Array,
  "skills": Array,
  "education": Array,
  "qualifications": Array,
  "experience": 11,
  "projects": Array,
  "accolades": 2,
  "contact_number": "0491570156",
  "contact_email": Object,
  "related_contacts": Array,
  "impression": "excellent"
}

{
  "_id": ObjectId("631d39ce1d896dc2e441044e"),
  "name": "Catarina Moreira",
  "meeting_place": Object,
  "age": 31,
  "organisation": Array,
  "skills": Array,
  "education": Array,
  "qualifications": Array,
  "experience": 6,
  "projects": Array,
  "accolades": 3,
  "contact_number": "0412550274",
  "contact_email": Object,
  "related_contacts": Array,
  "impression": "Excellent"
}
```

```
Atlas atlas-dr771w-shard-0 [primary] careerContacts> db.contactInfo.findOne({$and:[{"organisation.0.name": "/University/"}, {"organisation.0.position": "Lecturer"}, {"skills": "Data Analytics"}]})
{
  "_id": ObjectId("631d39da1d896dc2e4410438"),
  "name": "Emmeline Preece",
  "meeting_place": {
    "location": "QUT Gardens Point Campus",
    "date": "20/07/2021",
    "event": "Orientation"
  },
  "age": 33,
  "organisation": [
    {
      "name": "Queensland University of Technology",
      "position": "Lecturer"
    },
    {
      "name": "AIA Australia Limited",
      "position": "Data Analyst"
    }
  ],
  "skills": [ "Data Analytics", "Modern Data Management" ],
  "education": [ "Queensland University of Technology" ],
  "qualifications": [ "PHD in Data Science", "Bachelor of Computer Science" ],
  "experience": 11,
  "projects": [ "Voice Recognition", "Sound Recognition" ],
  "accolades": 2,
  "contact_number": "0491570156",
  "contact_email": { personal: "emmelineprece@gmail.com", work: "e.preece@qut.edu.au" },
  "related_contacts": [ "Zayne Bellamy", "Safiya Marsh" ],
  "impression": "excellent"
}
```

## careerContacts.contactInfo

30 1  
DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter: `{ $and: [{ "organisation.0.name": "/University/" }, { "organisation.0.position": "Lecturer" }, { "skills": "Data Analytics" } ] }` OPTIONS FIND RESET ↻ ⋮

PROJECT { field: 0 }

SORT { field: -1 } or [[field], -1]] MAX TIME MS 60000

COLLATION { locale: 'simple' } SKIP 0 LIMIT 1

ADD DATA VIEW 0

Displaying documents 1 - 1 of 1 < > C REFRESH

```
{
  "_id": ObjectId("631d39da1d896dc2e4410438"),
  "name": "Emmeline Preece",
  "meeting_place": Object,
  "age": 33,
  "organisation": Array,
  "skills": Array,
  "education": Array,
  "qualifications": Array,
  "experience": 11,
  "projects": Array,
  "accolades": 2,
  "contact_number": "0491570156",
  "contact_email": Object,
  "related_contacts": Array,
  "impression": "excellent"
}
```

## Question 5.

`db.contactInfo.findOne({$and:[{"skills": {$in: ["Modern Data Management", "Process Mining", "Automation"]}}, {"experience": {$gte: 2}}],{_id: 0, name: 1})`

Find the name of the first contact who has at least one of these skills: 'modern data management', 'process mining', and 'automation' and has greater or equal to 2 years of experience.

careerContacts.contactInfo 30 1  
DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

{\$and:[{"skills": {\$in: ["Modern Data Management", "Process Mining", "Automation"]}, {"experience": {\$gte: 2}}]}

Displaying documents 1 - 9 of 9

```
{
  "_id": ObjectId("631d39da1d896dc2e4410431"),
  "name": "Ravihansa Rajapakse",
  "meeting_place": Object,
  "age": 32,
  "organisation": Array,
  "skills": Array,
  "education": Array,
  "qualifications": Array,
  "experience": 19,
  "projects": Array,
  "accolades": 5,
  "contact_number": "0413014820",
  "contact_email": Object,
  "related_contacts": Array,
  "impression": "good"
}
```

```
{
  "_id": ObjectId("631d39da1d896dc2e4410438"),
  "name": "Emmeline Preece",
  "meeting_place": Object,
  "age": 33,
  "organisation": Array,
  "skills": Array,
  "education": Array,
  "qualifications": Array,
  "experience": 11,
  "projects": Array,
  "accolades": 2,
  "contact_number": "0491570156",
  "contact_email": Object,
  "related_contacts": Array,
  "impression": "excellent"
}
```

Atlas atlas-dn77lw-shard-0 [primary] careerContacts> db.contactInfo.findOne({'\$and':[{'skills': {'\$in': ['Modern Data Management', 'Process Mining', 'Automation']}}, {'experience': {'\$gte': 2}}]}, {'\_id': 0, 'name': 1})

careerContacts.contactInfo 30 1  
DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

{\$and:[{"skills": {\$in: ["Modern Data Management", "Process Mining", "Automation"]}, {"experience": {\$gte: 2}}]}

{ "\_id": 0, "name": 1 }

{ field: -1 } or [ { field, -1 } ]

{ locale: 'simple' }

0  1

Displaying documents 1 - 1 of 1

```
{
  "name": "Ravihansa Rajapakse"
}
```

## Question 6.

db.contactInfo.findOne({'\$and':[{"impression": "good"}, {"experience": {"\$gte": 10}}]}, {'\_id': 0, "name": 1})

Finds the name of the first contact with overall impression of 'good' and years of experience that is greater than or equal to 10.



careerContacts.contactInfo 30 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

**FILTER** `{ $and: [{"impression": "good"}, {"experience": { $gte: 10 }}]}` **OPTIONS** **FIND** **RESET** **REFRESH**

**ADD DATA** **VIEW** **0**

Displaying documents 1 - 5 of 5

```

{
  "_id": ObjectId("631d39da1d896dc2e4410431"),
  "name": "Ravihansa Rajapakse",
  "meeting_place": Object,
  "age": 32,
  "organisation": Array,
  "skills": Array,
  "education": Array,
  "qualifications": Array,
  "experience": 19,
  "projects": Array,
  "accolades": 5,
  "contact_number": "0413014820",
  "contact_email": Object,
  "related_contacts": Array,
  "impression": "good"
}

{
  "_id": ObjectId("631d39da1d896dc2e4410432"),
  "name": "Colin Fidge",
  "meeting_place": Object,
  "age": 38,
  "skills": Array,
  "education": Array,
  "qualifications": Array,
  "experience": 18,
  "projects": Array,
  "accolades": 9,
  "contact_number": "0402236434",
  "contact_email": Object,
  "related_contacts": Array,
  "impression": "good"
}

```

```
Atlas atlas-dr77lw-shard-0 [primary] careerContacts> db.contactInfo.findOne({ $and: [{"impression": "good"}, {"experience": { $gte: 10 }}]}, { _id: 0, "name": 1 })
{ name: "Ravihansa Rajapakse" }
```

careerContacts.contactInfo 30 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

**FILTER** `{ $and: [{"impression": "good"}, {"experience": { $gte: 10 }}]}` **OPTIONS** **FIND** **RESET** **REFRESH**

**PROJECT** `{ _id: 0, "name": 1 }`

**SORT** `{ field: -1 } or [ { field, -1 } ]` **MAX TIME MS** 60000

**COLLATION** `{ locale: "simple" }` **SKIP** 0 **LIMIT** 1

**VIEW** **0**

Displaying documents 1 - 1 of 1

```

{
  "name": "Ravihansa Rajapakse"
}

```

## Question 7\*.

**Irfan:** I am looking for a job as a Data Analyst and need to seek advice from a professional with several years of experience in the field. Therefore, I need the contact information of individuals with 'data analytics' skill, PhD, at least 10 years of experience, and an "excellent" overall impression.

```
db.contactInfo.find({ $and: [{"skills": "Data Analytics"}, {"qualifications": "/PhD/}, {"experience": { $gte: 10 }}, {"impression": "excellent"}]}, { _id: 0, name: 1, contact_number: 1, contact_email: 1 })
```

The query finds any contacts with 'data analytics' as a skill, a PhD in their qualifications, number of experience that is greater or equal to 10, and an 'excellent' impression and projects only the name, contact number, and contact email of the matching users

careerContacts.contactInfo 30 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER: `{ "$and": [{"skills": "Data Analytics"}, {"qualifications": "/PhD/}, {"experience": {"$gte": 10}}, {"impression": "excellent"}]}`
 PROJECT: `{ field: 0 }`
 SORT: `{ field: -1 } or [[ 'field', -1 ]]`
 COLLATION: `{ locale: 'simple' }`
 MAX TIME MS: 60000
 SKIP: 0 LIMIT: 0

ADD DATA VIEW

Displaying documents 1 - 1 of 1

```

{
  "_id": ObjectId("631d39da1d896dc2e4410438"),
  "name": "Emmeline Preece",
  "meeting_place": Object,
  "age": 33,
  "organisation": Array,
  "skills": Array,
  "education": Array,
  "qualifications": Array,
  "experience": 11,
  "projects": Array,
  "accolades": 2,
  "contact_number": "0491570156",
  "contact_email": Object,
  "related_contacts": Array,
  "impression": "excellent"
}

```

Atlas atlas-dr77lw-shard-0 [primary] careerContacts> db.contactInfo.find({\$and: [{"skills": "Data Analytics"}, {"qualifications": "/PhD/}, {"experience": {"\$gte": 10}}, {"impression": "excellent"}]}, {\_id: 0, name: 1, contact\_number: 1, contact\_email: 1})

```

[
  {
    name: 'Emmeline Preece',
    contact_number: '0491570156',
    contact_email: {
      personal: 'emmelineprece@gmail.com',
      work: 'e.preece@qut.edu.au'
    }
  }
]

```

careerContacts.contactInfo 30 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER: `{ "$and": [{"skills": "Data Analytics"}, {"qualifications": "/PhD/}, {"experience": {"$gte": 10}}, {"impression": "excellent"}]}`
 PROJECT: `{ _id: 0, name: 1, contact_number: 1, contact_email: 1 }`
 SORT: `{ field: -1 } or [[ 'field', -1 ]]`
 COLLATION: `{ locale: 'simple' }`
 MAX TIME MS: 60000
 SKIP: 0 LIMIT: 0

VIEW

Displaying documents 1 - 1 of 1

```

name: "Emmeline Preece"
contact_number: "0491570156"
contact_email: Object
  personal: "emmelineprece@gmail.com"
  work: "e.preece@qut.edu.au"

```

**Nicholas:** I am searching for a contact who has the a lot of experience in the industry (~10). However, I only want to display their personal email only, and the education institutions they've attended.

```
db.contactInfo.find({"experience":{"$gte":10}}, {_id: 0, "name": 1, "contact_email.personal": 1, "education": 1})
```

Finds all documents with experience greater than or equal to 10. Prints name, personal email and education.

<div><div>_id: Object2d('631d39da1d896dc2e4410431')</div><div>Name: "Ravithana Rajapakse"</div><div>&gt; meeting_place: Object</div><div>age: 32</div><div>&gt; organisation: Array</div><div>&gt; skills: Array</div><div>&gt; education: Array</div><div>&gt; qualifications: Array</div><div>experience: 19</div><div>&gt; projects: Array</div><div>accolades: 5</div><div>contact_number: "0413014820"</div><div>&gt; contact_email: Object</div><div>&gt; related_contacts: Array</div><div>impression: "good"</div></div>
<div><div>_id: Object2d('631d39da1d896dc2e4410432')</div><div>Name: "Colin Fidge"</div><div>&gt; meeting_place: Object</div><div>age: 38</div><div>&gt; skills: Array</div><div>&gt; organisation: Array</div><div>&gt; education: Array</div><div>&gt; qualifications: Array</div><div>experience: 19</div><div>&gt; projects: Array</div><div>accolades: 5</div><div>contact_number: "0402236434"</div><div>&gt; contact_email: Object</div><div>&gt; related_contacts: Array</div><div>impression: "good"</div></div>

```

Atlas atlas-dr77lw-shard-0 [primary] careerContacts> db.contactInfo.find({"experience":{"$gte:10}}, {_id: 0, "name": 1, "contact_email.personal": 1, "education": 1})
[
  {
    name: 'Ravihansa Rajapakse',
    education: [ 'OUT', 'Perform Australia', 'University of Columbo' ],
    contact_email: {}
  },
  {
    name: 'Colin Fidge',
    education: [ 'Queensland University of Technology, University of Queensland' ],
    contact_email: {}
  },
  {
    name: 'Wade Milne',
    education: [ 'Queensland University of Technology' ],
    contact_email: { personal: 'wademilne@hotmail.com' }
  },
  {
    name: 'Leonie Simpson',
    education: [ 'Queensland University of Technology' ],
    contact_email: {}
  },
  {
    name: 'Kanika Goel',
    education: [ 'Queensland University of Technology' ],
    contact_email: {}
  },
  {
    name: 'Mark Broadmeadow',
    education: [
      'University of Southern Queensland',
      'Queensland University of Technology',
      'University of Queensland'
    ],
    contact_email: {}
  },
  {
    name: 'Emmeline Preece',
    education: [ 'Queensland University of Technology' ],
    contact_email: { personal: 'emmelineprece@gmail.com' }
  },
  {
    name: 'Oscar Aurora',
    education: [
      'University of Queensland',
      'Queensland University of Technology'
    ],
    contact_email: { personal: 'auroraoscar@gmail.com' }
  },
  {
    name: 'Frances Archer',
    education: [
      'University of Queensland',
      'Queensland University of Technology'
    ],
    contact_email: { personal: 'francesarcher@gmail.com' }
  },
]

```

```

[
  {
    name: 'Mason Baker',
    education: [ 'Griffith University', 'Monash University' ],
    contact_email: { personal: 'mason_baker@gmail.com' }
  },
  {
    name: 'Dakota Baldwin',
    education: [ 'University of New South Wales' ],
    contact_email: { personal: 'dakota.baldwin@outlook.com' }
  },
  {
    name: 'David Lovell',
    education: [ 'Queensland University of Technology, University of Queensland' ],
    contact_email: { personal: 'davel@gmail.com' }
  },
  {
    name: 'Patric Johnson',
    education: [
      'University of Technology Sydney',
      'Queensland University of Technology'
    ],
    contact_email: { personal: 'patricjohnson@gmail.com' }
  },
  {
    name: 'Henry Lowel',
    education: [ 'University of Queensland' ],
    contact_email: { personal: 'HenryLowee@gmail.com' }
  }
]

```

Documents Aggregations Schema Explain Plan Indexes Validation

**FILTER** `{ "experience": { "$gte": 10 } }` **OPTIONS** **FIND** **RESET** **...**

**PROJECT** `{ "_id": 0, "name": 1, "contact_email.personal": 1, "education": 1 }`

**SORT** `{ "field": -1 } or [{"field": -1}]` **MAX TIME MS** 60000

**COLLATION** `{ locale: 'simple' }` **SKIP** 0 **LIMIT** 0

VIEW **JSON** **YAML** **CSV**

Displaying documents 1 - 14 of 14 **REFRESH**

```
name: "Ravihansa Rajapakse"
> education: Array
> contact_email: Object

name: "Colin Fudge"
> education: Array
> contact_email: Object

name: "Wade Milne"
> education: Array
> contact_email: Object

name: "Leslie Simpson"
> education: Array
> contact_email: Object

name: "Kanika Goel"
> education: Array
> contact_email: Object

name: "Mark Broadmeadow"
> education: Array
> contact_email: Object
```

**Andrew:** I am interested in participating in research, particularly in the fields of cryptography or data analytics. As such, I should query the database to extract the relevant information I need to email all current nearby university lecturers who have experience in these particular fields and enquire about any potential opportunities.

```
db.contactInfo.find({"organisation.0.position":
"Lecturer", $or:[{"skills":"Cryptography"}, {"skills":"Data
Analytics"}]}), { _id: 0, "name": 1, "organisation": { "$slice": 1 }, "skills": 1, "projects": 1, "contact_email.work": 1 })
```

```

Atlas atlas-dr77lw-shard-0 [primary] careerContacts> db.contactInfo.find({'organisation.0.position': 'Lecturer', $or:[{'skills':'Cryptography'},
{'skills':'Data Analytics'}]}), {_id:0, 'name':1, 'organisation':{$slice:1}, 'skills':1, 'projects':1, 'contact_email.work':1})
[
  {
    name: 'Leonie Simpson',
    organisation: [
      {
        name: 'Queensland University of Technology',
        position: 'Lecturer'
      }
    ],
    skills: [ 'Cryptography', 'Cryptanalysis' ],
    projects: [
      'Was a member of a group that completed bug bounties at a large scale'
    ],
    contact_email: { work: 'lr.simpson@qut.edu.au' }
  },
  {
    name: 'Emmeline Preece',
    organisation: [
      {
        name: 'Queensland University of Technology',
        position: 'Lecturer'
      }
    ],
    skills: [ 'Data Analytics', 'Modern Data Management' ],
    projects: [ 'Voice Recognition', 'Sound Recognition' ],
    contact_email: { work: 'e.preece@qut.edu.au' }
  },
  {
    name: 'Catarina Moreira',
    organisation: [
      {
        name: 'Queensland University of Technology',
        position: 'Lecturer'
      }
    ],
    skills: [ 'Modern Data Management', 'Data Analytics' ],
    projects: [ 'Sentiment Analysis', 'Leaf Disease Detection' ],
    contact_email: { work: 'catarina.pintomoreira@qut.edu.au' }
  }
]

```

Finds all documents where the current organisation's position is 'Lecturer', and has skills of either cryptography or data analytics. Prints name, current organisation, skills, projects, and work email.

**Minjae Lee:** While I was searching, I mainly looked for people in the same field as me. For instance, computer science, mathematics, engineering, and things about data. I searched SNS, the Internet, or school sites, and everywhere to find their information, and also contacts, such as email, and phone number.

```

db.contactInfo.find({"skills": { $in: ["Data Science", "Modern Data Management",
"Programming", "Computer Science", "Mathematics"]}}, {_id: 0, "name":1,
"contact_number": 1, "contact_email": 1})

```

Finds all documents where skills have "Data Science", "Modern Data Management", "Programming", "Computer Science", "Mathematics".

FILTER

`{ "skills": { $in: ["Data Science", "Modern Data Management", "Programming", "Computer Science", "Mathematics"] }}`

OPTIONS

FIND RESET ↺ ..

PROJECT

`{ field: 0 }`

SORT

`{ field: -1 } or [[ 'field', -1 ]]`

MAX TIME MS 60000

COLLATION

`{ locale: 'simple' }`

Skip 0 Limit 0

ADD DATA ▾

VIEW

Displaying documents 1 - 20 of 22 < > C REFRESH

```
{
  _id: ObjectId("631d39da1d996dc2e4410431"),
  name: "Revihansha Rajapakse",
  meeting_place: Object,
  age: 32,
  organisation: Array,
  skills: Array,
  education: Array,
  qualifications: Array,
  experience: 19,
  projects: Array,
  accolades: 5,
  contact_number: "0413014820",
  contact_email: Object,
  related_contacts: Array,
  impression: "good"
}
```

```

_id: ObjectId('631d93da1d96dc2e4410432')
name: "Colin Fidge"
meeting_place: Object
age: 35
skills: Array
organisation: Array
education: Array
qualifications: Array
experience: 18
projects: Array
accolades: 9
contact_number: "0402236434"
contact_email: Object
related_contacts: Array
impression: "good"

```

```

cat stlts-d-772e-shrd-0 [primary] career:contacts db:contacts.find({'_id': {'$in': ['Data Science', 'Machine Data Management', 'Programming', 'Computer Science', 'Mathematics']}}, {'_id': 0, 'name': 1, 'contact_email': 1})
{
  name: 'Ravihansa Rajapaksa',
  contact_number: '049178454',
  contact_email: { work: 'r.sampa.rajpaksap@qut.edu.au' }
}
{
  name: 'Colin Fidge',
  contact_number: '0492236434',
  contact_email: { work: 'c.fidge@qut.edu.au' }
}
{
  name: 'Emmeline Pryce',
  contact_number: '0491770150',
  contact_email: {
    personal: 'emmelinepryce@gmail.com',
    work: 'e.pryce@qut.edu.au'
  }
}
{
  name: 'Ezra Bellamy',
  contact_number: '0491770157',
  contact_email: {
    personal: 'ezrabellamy@gmail.com',
    work: 'e.bellamy@qut.edu.au'
  }
}
{
  name: 'Sallya Marsh',
  contact_number: '0491770150',
  contact_email: { personal: 'sallyamash@gmail.com', work: 's.marsh@qut.edu.au' }
}
{
  name: 'Freddy Kaiser',
  contact_number: '0491770150',
  contact_email: {
    personal: 'freddykaiser@gmail.com',
    work: 'f.kaiser@trattontudiosgames.com'
  }
}
{
  name: 'Oscar Aurora',
  contact_number: '0491770160',
  contact_email: { personal: 'auroraoscar@gmail.com', work: 'o.aurora@tracoo.com' }
}
{
  name: 'Frances Archer',
  contact_number: '0491770161',
  contact_email: {
    personal: 'francesarcher@gmail.com',
    work: 'f.archer@qut.edu.au'
  }
}

```

```
{
  name: 'Nanette Harvey',
  contact_number: '041379102',
  contact_email: {
    personal: 'nanetteharvey@gmail.com',
    work: 'n.harvey@digitalpeople.com.au'
  }
},
{
  name: 'Mason Baker',
  contact_number: '0455997817',
  contact_email: {
    personal: 'mason_baker@gmail.com',
    work: 'm.baker@treasury.qld.gov.au'
  }
},
{
  name: 'Daiota Baladin',
  contact_number: '0414419339',
  contact_email: {
    personal: 'daiota.baladin@outlook.com',
    work: 'dbaladin14@eq.edu.au'
  }
},
{
  name: 'Tahlina Rodgers',
  contact_number: '0487071143',
  contact_email: { personal: 't.rodgers@hotmail.com', work: 't.rodgers@eq.edu.au' }
},
{
  name: 'Bradley Cooper',
  contact_number: '0412242326',
  contact_email: {
    personal: 'brad.cooper@gmail.com',
    work: 'bradleycooper@igna.com'
  }
},
{
  name: 'Stephen Hudson',
  contact_number: '0412182181',
  contact_email: {
    personal: 'stephen.hudson@outlook.com',
    work: 'stephen.h@boeing.com.au'
  }
},
{
  name: 'Elliot Smith',
  contact_email: { work: 'e.smith@hp.com.au' }
},
{
  name: 'David Lovell',
  contact_number: '0414229493',
  contact_email: { personal: 'daveil@gmail.com', work: 'david.lovell@out.edu.au' }
},
{
  name: 'Lowel Johnson',
  contact_number: '0491222321',
  contact_email: { personal: 'lowee@hotmail.com', work: 'loweljohnson@gmail.com' }
},
}
```

```
{
  name: 'Patricia Stewen',
  contact_number: '0441194923',
  contact_email: {
    personal: 'patystew@hotmail.com',
    work: 'patstewen@gmail.com'
  }
},
{
  name: 'Patrick Johnson',
  contact_number: '0412449582',
  contact_email: {
    personal: 'patrickjohnson@gmail.com',
    work: 'p.johnson@trattatstudios.com'
  }
},
{
  name: 'Ryan Heneghan',
  contact_number: '0489629572',
  contact_email: { personal: 'r.yhen@gmail.com', work: 'r.heneghan@out.edu.au' }
},
}
```

Type "it" for more  
Atlas atlas-dr77lw-shard-0 [primary] careerContacts

## careerContacts.contactInfo

30 1  
DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

**FILTER** { "skills": { \$in: ["Data Science", "Modern Data Management", "Programming", "Computer Science", "Mathematics"] } } **OPTIONS** **FIND** **RESET** **...**

**PROJECT** { "\_id": 0, "name": 1, "contact\_number": 1, "contact\_email": 1 }

**SORT** { field: -1 } or [ [ 'field', -1 ] ]

**COLLATION** { locale: 'simple' }

**SKIP** 0

**MAX TIME MS** 60000

**LIMIT** 0

**VIEW** **...** **...**

Displaying documents 1 - 20 of 22 **REFRESH**

name: "Devihansa Rajapakse"  
contact\_number: "0413014820"  
> contact\_email: Object

name: "Colin Fidge"  
contact\_number: "0402236434"  
> contact\_email: Object

name: "Emeline Freese"  
contact\_number: "0491570156"  
> contact\_email: Object

name: "Zayne Bellamy"  
contact\_number: "0491570157"  
> contact\_email: Object

name: "Safiya Marsh"  
contact\_number: "0491570158"  
> contact\_email: Object

name: "Freddy Kaiser"  
contact\_number: "0491570159"  
> contact\_email: Object



## Update:

### Question 8.

`db.contactInfo.updateOne({"name": "Kanika Goel"},{$set:{"impression": "good"}})`

Change the overall impression to 'good' for the first contact with the name 'Kanika Goel'.

careerContacts.contactInfo 30 1  
DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Displaying documents 1 - 1 of 1

```
{
  "_id": ObjectId("631d39da1d996dc2e4410435"),
  "name": "Kanika Goel",
  "meeting_place": Object,
  "age": 29,
  "organisation": Array,
  "skills": Array,
  "education": Array,
  "qualifications": Array,
  "experience": 12,
  "projects": Array,
  "accomplishes": 2,
  "contact_number": "04731394780",
  "contact_email": Object,
  "related_contacts": Array,
  "impression": "average"
}
```

Atlas atlas-dr77lw-shard-0 [primary] careerContacts> db.contactInfo.updateOne({"name": "Kanika Goel"},{\$set:{"impression": "good"}})

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

careerContacts.contactInfo 30 1  
DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Displaying documents 1 - 1 of 1

```
{
  "name": "Kanika Goel",
  "impression": "good"
}
```

## Question 9.

```
db.contactInfo.updateOne({"organisation.0.name": "Activision Blizzard"},{$inc: {"experience":2}})
```

Increment the number of experience by 2 years for a contact with Activision Blizzard as the current organisation.

The screenshot shows the MongoDB Compass interface. A filter is applied: `{"organisation.name": "Activision Blizzard"}`. The document view shows the following fields:

- `_id`: ObjectId('631d3cceld896dc2e4410447')
- `name`: "Kalani Sukhdeep"
- `meeting_place`: Object
- `organisation`: Array
  - `0`: Object
    - `name`: "Activision Blizzard"
    - `position`: "Director"
  - `1`: Object
- `skills`: Array
- `education`: Array
- `qualifications`: Array
- `experience`: 7
- `contact_number`: "0402391057"
- `contact_email`: Object
- `related_contacts`: Array
- `impression`: "good"

```
Atlas atlas-elm06o-shard-0 [primary] careerContacts> db.contactInfo.updateOne({"organisation.0.name": "Activision Blizzard"},{$inc: {"experience":2}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

The screenshot shows the MongoDB Compass interface for the `careerContacts.contactInfo` collection. The filter is `{"organisation.name": "Activision Blizzard"}`. The document view shows the following fields:

- `name`: "Kalani Sukhdeep"
- `experience`: 9

The interface also shows 30 documents and 1 index.

### Question 10.

```
db.contactInfo.updateOne({"name": "Ravihansa  
Rajapakse"},{$set:{"meeting_place.location": "QUT Kelvin Grove","meeting_place.date":  
"02/11/2021", "meeting_place.time": "8:00"}})
```

Set the meeting place location to QUT Kelvin Grove, date to 02/11/2021, time to 8.00 for a contact with the name 'Ravihansa Rajapakse'.

careerContacts.contactInfo

DOCUMENTSINDEXES

DocumentsAggregationsSchemaExplain PlanIndexesValidation

FILTER{"name": "Ravihansa Rajapakse"}

PROJECT{\_id@, name@, meeting\_place: @}

SORT{ field: -@ } or [{"field", -@}]

COLLATION{ locale: "simple" }

MAX TIME MS60000

SKIP0LIMIT0

VIEWS

Displaying documents 1 - 1 of 1

C REFRESH

name: "Ravihansa Rajapakse"

meeting\_place: Object

location: "QUT Gardens Point Campus"

date: "04/08/2022"

Atlas atlas-dr77lu-shard-0 [primary] careerContacts> db.contactInfo.updateOne({"name": "Ravihansa Rajapakse"},{\$set:{"meeting\_place.location": "QUT Kelvin Grove","meeting\_place.date": "02/11/2021","meeting\_place.time": "8:00"}})

{ acknowledged: true, insertedId: null, matchedCount: 1, modifiedCount: 1, upsertedCount: 0 }

Atlas atlas-dr77lu-shard-0 [primary] careerContacts>

careerContacts.contactInfo

30DOCUMENTS1INDEXES

DocumentsAggregationsSchemaExplain PlanIndexesValidation

FILTER{"name": "Ravihansa Rajapakse"}

PROJECT{\_id@, name@, meeting\_place: @}

SORT{ field: -@ } or [{"field", -@}]

COLLATION{ locale: "simple" }

MAX TIME MS60000

SKIP0LIMIT0

VIEWS

Displaying documents 1 - 1 of 1

C REFRESH

name: "Ravihansa Rajapakse"

meeting\_place: Object

location: "QUT Kelvin Grove"

date: "02/11/2021"

time: "8:00"

### Question 11.

```
db.contactInfo.updateOne({name:"Zayne Bellamy", skills:
"Databases"},{$set:{"skills.$":"Modern Databases"}})
```

Updates an array, 'skills', of contact with the name 'Zayne Bellamy' from 'Databases' to 'Modern Databases'

## careerContacts.contactInfo

30 1  
DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER {name:"Zayne Bellamy"}  
 PROJECT {\_id:0, name:1, skills:1}  
 SORT { field: -1 } or [{"field", -1}]  
 COLLATION { locale: "simple" }

MAX TIME MS 60000  
SKIP 0 LIMIT 0

VIEW

Displaying documents 1 - 1 of 1

```

name: "Zayne Bellamy"
skills: Array
  0: "Modern Data Management"
  1: "Databases"
  
```

```

Atlas atlas-dr77lw-shard-0 [primary] careerContacts> db.contactInfo.updateOne({name:"Zayne Bellamy", skills: "Databases"},{$set:{"skills.$":"Modern Databases"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
  
```

## careerContacts.contactInfo

30 1  
DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER {name:"Zayne Bellamy"}  
 PROJECT {\_id:0, name:1, skills:1}  
 SORT { field: -1 } or [{"field", -1}]  
 COLLATION { locale: "simple" }

MAX TIME MS 60000  
SKIP 0 LIMIT 0

VIEW

Displaying documents 1 - 1 of 1

```

name: "Zayne Bellamy"
skills: Array
  0: "Modern Data Management"
  1: "Modern Databases"
  
```

## Question 12.

```
db.contactInfo.updateOne({"organisation.0.name": "Queensland University of Technology"},
{$addToSet:{"skills": "Automation"}})
```

Add 'automation' to the list of skills for a contact with 'Queensland University of Technology' as the current organisation.

## careerContacts.contactInfo

30 1  
DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER {"organisation.0.name": "Queensland University of Technology"}  
 PROJECT {\_id:0, name:1, organisation:1, skills:1}  
 SORT { field: -1 } or [{"field", -1}]  
 COLLATION { locale: "simple" }

MAX TIME MS 60000  
SKIP 0 LIMIT 1

VIEW

Displaying documents 1 - 1 of 1

```

name: "Colin Fidge"
skills: Array
  0: "Computer Science"
  1: "Programming"
organisation: Array
  0: Object
    name: "Queensland University of Technology"
    position: "Unit Coordinator"
  
```

```

Atlas atlas-dr77lw-shard-0 [primary] careerContacts> db.contactInfo.updateOne({"organisation.0.name": "Queensland University of Technology"}, {$addToSet:{"skills": "Automation"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
  
```

careerContacts.contactInfo

301

DOCUMENTSINDEXES

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

FILTER

{ "name": "Freddy Kaiser" }

OPTIONS

PROJECT

{ "\_id": 0, name: 1, organisation: 1, "contact\_email.work": 1 }

SORT

{ field: -1 } or [{ 'field', -1 }]

MAX TIME MS60000

COLLATION

{ locale: 'simple' }

SKIP0

LIMIT1

VIEW

Displaying documents 1 - 1 of 1

REFRESH

```

name: "Freddy Kaiser"
organisation: Array
  - 0: Object
    name: "Stratton Studios"
    position: "Unity Developer"
contact_email: Object
  work: "f.kaiser@strattonstudiosgames.com"

```

```

atlas atlas-d771w-shard-0 [primary] careerContacts> db.contactInfo.updateOne({ "name": "Freddy Kaiser" }, { $set: { "contact_email.work": "f.kaiser@whitebay.com.au" }, $push: { "organisation": { $each: [{ "name": "White Bay Search", "position": "Game Designer" }] } } })
acknowledged: true,
insertedId: null,
matchedCount: 1,
modifiedCount: 1,
upsertedCount: 0

```

careerContacts.contactInfo

30 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

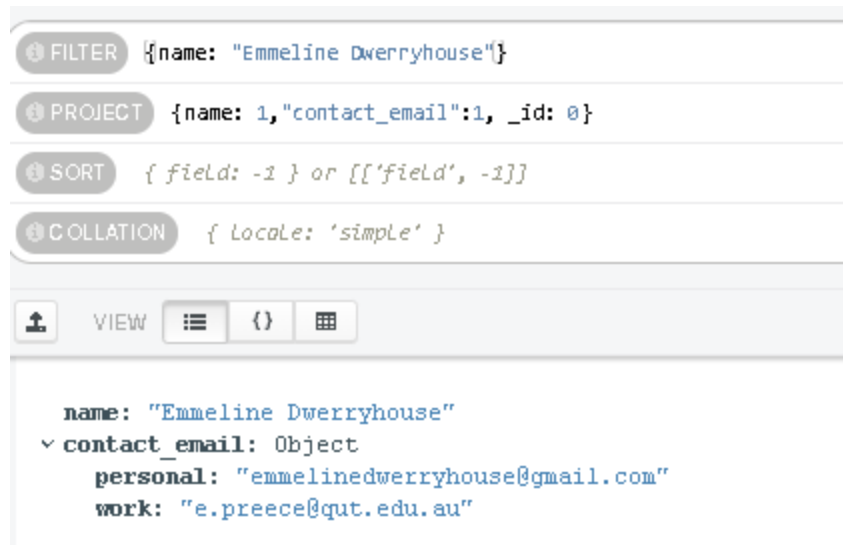
**FILTER** { "name": "Freddy Kaiser" } **OPTIONS** **FIND** **RESET**

**PROJECT** { \_id: 0, name: 1, organisation: 1, "contact\_email.work": 1 }

**SORT** { field: -1 } or [ [ 'field', -1 ] ] **MAX TIME MS** 68808

**COLLATION** { locale: 'simple' } **SKIP** 0 **LIMIT** 1

VIEW **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T** **U** **V** **W** **X** **Y** **Z** **0** **1** **2** **3** **4** **5** **6** **7** **8** **9** **A** **B** **C** **D** **E** **F** **G** **H** **I** **J** **K** **L** **M** **N** **O** **P** **Q** **R** **S** **T**



**Andrew:** While chatting with Bradley, he mentioned that he recently completed his graduate diploma in interaction design at UQ. As such, I need to update his skills, education and qualifications fields in the database.

```
db.contactInfo.updateOne({"name":"Bradley Cooper"},{$push:{"skills":"UX Design","education":"University of Queensland","qualifications":"Graduate Diploma in Interaction Design"}})
```

**FILTER** `{ "name": "Bradley Cooper" }`

**PROJECT** `{ _id:0, name:1, skills:1, qualifications:1, education:1 }`

**SORT** `{ field: -1 } or [[ 'field', -1 ]]`

**COLLATION** `{ locale: 'simple' }`



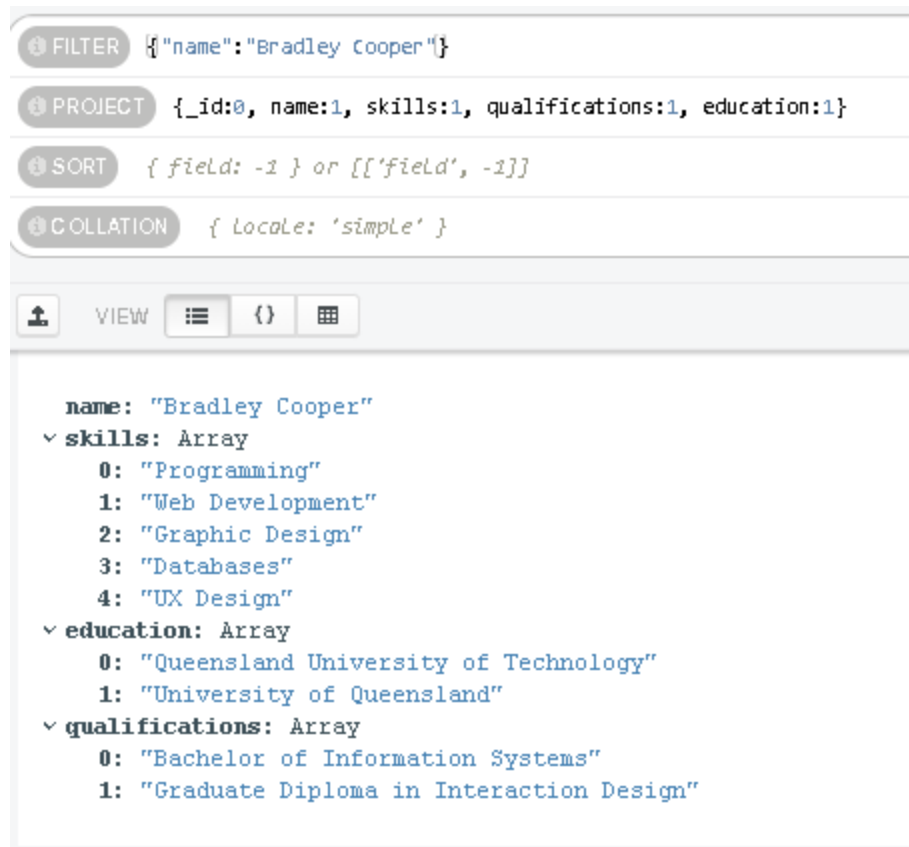
VIEW



```
name: "Bradley Cooper"
skills: Array
  0: "Programming"
  1: "Web Development"
  2: "Graphic Design"
  3: "Databases"
education: Array
  0: "Queensland University of Technology"
qualifications: Array
  0: "Bachelor of Information Systems"
```

```
Atlas atlas-dr77lw-shard-0 [primary] careerContacts: db.contactInfo.updateOne({ "name": "Bradley Cooper" }, { $push: { "skills": "UX Design", "education": "University of Queensland", "qualifications": "Graduate Diploma in Interaction Design" } },
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
})
```





**Minjae Lee:** Recently, I got in touch with Henry Lowel again. I asked him if this information is correct, which I used for this database. He said he moved the company to Google, but the position is the same. Also he recently finished his bachelor degree in mathematics in Queensland University of Technology.

```
db.contactInfo.updateOne({"name":"Henry Lowel"},{$set:{"organisation.0.name":"Google"},$push:{"skills":"Mathematics","education":"Queensland University of Technology","qualifications":"Bachelor of Mathematics"}})
```

**FILTER** `{ "name": "Henry Lowel" }` **OPTIONS** **FIND** **RESET** **↺** **⋮**

**PROJECT** `0, name:1, skills: 1, organisation: 1, education: 1, qualifications: 1`

**SORT** `{ field: -1 } or [['field', -1]]` **MAX TIME MS** 60000

**COLLATION** `{ locale: 'simple' }` **SKIP** 0 **LIMIT** 0

**VIEW** **⌵** **{ }** **⌶** Displaying documents 1 - 1 of 1 **<** **>** **REFRESH**

```

name: "Henry Lowel"
organisation: Array
  0: Object
    name: "Canva"
    position: "Software Engineering"
skills: Array
  0: "Computer Science"
education: Array
  0: "University of Queensland"
qualifications: Array
  0: "Bachelor of Computer Science"
  1: "Phd in Computer Science"

```

```

Atlas atlas-dr77lw-shard-0 [primary] careerContacts> db.contactInfo.updateOne({ "name": "Henry Lowel" }, { $set: { organisation: { _id: 0, name: "Google" }, $push: { skills: "Mathematics", education: "Queensland University of Technology", qualifications: "Bachelor of Mathematics" } } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
Atlas atlas-dr77lw-shard-0 [primary] careerContacts>

```

**FILTER** `{ "name": "Henry Lowel" }`

**PROJECT** `{ _id: 0, name: 1, organisation: 1, skills: 1, education: 1, qualifications: 1 }`

**SORT** `{ field: -1 } or [['field', -1]]`

**COLLATION** `{ locale: 'simple' }` **SKIP**

**VIEW** **⌵** **{ }** **⌶**

```

name: "Henry Lowel"
organisation: Array
  0: Object
    name: "Google"
    position: "Software Engineering"
skills: Array
  0: "Computer Science"
  1: "Mathematics"
education: Array
  0: "University of Queensland"
  1: "Queensland University of Technology"
qualifications: Array
  0: "Bachelor of Computer Science"
  1: "Phd in Computer Science"
  2: "Bachelor of Mathematics"

```

## Delete:

### Question 14.

```
db.contactInfo.deleteOne({"experience":{$lte:2}})
```

Deletes the first document with experience less than or equal to 2

**careerContacts.contactInfo** 30 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

**FILTER** {"experience":{\$lte:2}} **OPTIONS** **FIND** **RESET** **↺** **...**

**PROJECT** {\_id: 0, name: 1, experience: 1}

**SORT** { field: -1 } or [['field', -1]] **MAX TIME MS** 60000

**COLLATION** { locale: 'simple' } **SKIP** 0 **LIMIT** 0

**VIEW** **{} {} {}** Displaying documents 1 - 4 of 4 **REFRESH**

```
name: "Zayne Bellamy"
experience: 2
```

```
name: "Nanette Harvey"
experience: 2
```

```
name: "Bradley Cooper"
experience: 2
```

```
name: "Patson Steven"
experience: 1
```

```
Atlas atlas-dr77lw-shard-0 [primary] careerContacts> db.contactInfo.deleteOne({"experience":{$lte:2}})
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-dr77lw-shard-0 [primary] careerContacts>
```

FILTER {"experience":{"\$lte:2}}

OPTIONS

FIND

RESET



PROJECT {\_id: 0, name: 1, experience: 1}

SORT { field: -1 } or [['field', -1]]

MAX TIME MS 60000

COLLATION { locale: 'simple' }

SKIP 0

LIMIT 0



VIEW



Displaying documents 1 - 3 of 3



REFRESH

```
name: "Nanette Harvey"
experience: 2
```

```
name: "Bradley Cooper"
experience: 2
```

```
name: "Patson Steven"
experience: 1
```

### Question 15.

```
db.contactInfo.deleteMany({"organisation.name":"Queensland University of Technology"})
```

Deletes as many documents with the organisation name of 'Queensland University of Technology', former or current.

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

FILTER {"organisation.name":"Queensland University of Technology"} OPTIONS

FIND

RESET



PROJECT {\_id: 0, name: 1, organisation: 1}

SORT { field: -1 } or [['field', -1]]

MAX TIME MS 60000

COLLATION { locale: 'simple' }

SKIP 0

LIMIT 0



VIEW



Displaying documents 1 - 12 of 12



REFRESH

```
name: "Colin Fidge"
organisation: Array
  0: Object
    name: "Queensland University of Technology"
    position: "Unit Coordinator"
```

```
name: "Wade Milne"
organisation: Array
  0: Object
  1: Object
    name: "Queensland University of Technology"
    position: "Lecturer"
```

```
name: "Leonie Simpson"
organisation: Array
  0: Object
    name: "Queensland University of Technology"
    position: "Lecturer"
```

```
name: "Kanika Goel"
organisation: Array
  0: Object
    name: "Queensland University of Technology"
    position: "Lecturer"
```

```
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-dr77lw-shard-0 [primary] careerContacts> db.contactInfo.deleteMany({"organisation.name":"Queensland University of Technology"})
{ acknowledged: true, deletedCount: 12 }
```

FILTER

{ "organisation.name": "Queensland University of Technology" }

OPTIONS

PROJECT

{ \_id: 0, name: 1, organisation: 1 }

SORT

{ field: -1 } or [[ 'field', -1 ]]

MAX TIME MS

60000

COLLATION

{ locale: 'simple' }

SKIP

0

LIMIT

0

VIEW

Displaying documents 0 - 0 of 0

<

>

REFRESH

No results

Try to modify your query to get results

## Part B:

### Question 16.

My friends and I have decided to have a get-together at my house. To accommodate them, I have bought various food products. In case I run out of food, I'll want to keep these products in a secure, and accessible database for easy reference, and so, I'll use MongoDB to help me manage my food inventory.



## Create:









### Document Structure:


name	Name of the product	String
types	List of product types	Array of String
purchase_date	Date of purchase	Document type[day (integer), month (integer), year (integer)]
expiry_date	Expiration date	Document type[day (integer), month (integer), year (integer)]
location	The location where the product is stored	String
Amount	Description of the quantity	String

1. I want to insert some products that I have bought into the database.

```
([{name: "Full Cream Milk", types: ["Dairy", "Milk"], purchase_date: {day: 28, month: 8, year: 2022}, expiry_date: {day: 2, month: 9, year: 2022}, location: "fridge", amount: "700 ml"},{name: "Chunk Tuna With Chilli", types: ["Canned","Seafood", "Fish"], purchase_date: {day: 14, month: 7, year: 2022}, expiry_date: {day: 10, month: 1, year: 2024},location: "pantry", amount: "180 g"},{name: "Broccoli", types: ["Vegetable"], purchase_date: {day: 1, month: 9, year: 2022}, expiry_date: {day: 5, month: 9, year: 2022}, location: "fridge", amount: "180 g"},{name: "Chicken Breast", types: ["Meat", "Chicken"], purchase_date: {day: 1, month: 9, year: 2022}, expiry_date: {day: 1, month: 6, year: 2023}, location: "freezer", amount: "250 g"},{name: "Thickened Cream", types: ["Dairy", "Cream"], purchase_date: {day: 22, month: 8, year: 2022}, expiry_date: {day: 28, month: 8, year: 2022}, location: "fridge", amount: "200 ml"},{name: "Ground Beef", types: ["Meat", "Beef"], purchase_date: {day: 1, month: 9, year: 2022}, expiry_date: {day: 1, month: 1, year: 2023}, location: "freezer", amount: "500 g"},{name: "Oreo", types: ["Snack", "Cookie"], purchase_date: {day: 22, month: 8, year: 2022}, expiry_date: {day: 6, month: 4, year: 2023}, location: "pantry", amount: "12"}])
```

**FILTER** { field: 'value' } **OPTIONS** **FIND** **RESET**  

 **ADD DATA**  **VIEW**    Displaying documents 0 - 0 of 0    **REFRESH**



## This collection has no data

It only takes a few seconds to import data from a JSON or CSV file

**Import Data**

```
Atlas atlas-dr77lw-shard-0 [primary] Inventory> db.foodInventory.insertMany([{"name": "Full Cream Milk", "types": ["Dairy", "Milk"], "purchase_date": {"day": 28, "month": 8, "year": 2022}, "expiry_date": {"day": 2, "month": 9, "year": 2022}, "location": "fridge", "amount": "700 ml"}, {"name": "Chunk Tuna With Chilli", "types": ["Canned", "Seafood", "Fish"], "purchase_date": {"day": 14, "month": 7, "year": 2022}, "expiry_date": {"day": 10, "month": 1, "year": 2024}, "location": "pantry", "amount": "180 g"}, {"name": "Broccoli", "types": ["Vegetable"], "purchase_date": {"day": 1, "month": 9, "year": 2022}, "expiry_date": {"day": 5, "month": 9, "year": 2022}, "location": "fridge", "amount": "180 g"}, {"name": "Chicken Breast", "types": ["Meat", "Chicken"], "purchase_date": {"day": 1, "month": 9, "year": 2022}, "expiry_date": {"day": 1, "month": 6, "year": 2023}, "location": "freezer", "amount": "250 g"}, {"name": "Thickened Cream", "types": ["Dairy", "Cream"], "purchase_date": {"day": 22, "month": 8, "year": 2022}, "expiry_date": {"day": 28, "month": 8, "year": 2022}, "location": "fridge", "amount": "200 ml"}, {"name": "Ground Beef", "types": ["Meat", "Beef"], "purchase_date": {"day": 1, "month": 9, "year": 2022}, "expiry_date": {"day": 1, "month": 1, "year": 2023}, "location": "freezer", "amount": "500 g"}, {"name": "Oreo", "types": ["Snack", "Cookie"], "purchase_date": {"day": 22, "month": 8, "year": 2022}, "expiry_date": {"day": 6, "month": 4, "year": 2023}, "location": "pantry", "amount": "12"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("631c20b37649348dfa54814b"),
    '1': ObjectId("631c20b37649348dfa54814c"),
    '2': ObjectId("631c20b37649348dfa54814d"),
    '3': ObjectId("631c20b37649348dfa54814e"),
    '4': ObjectId("631c20b37649348dfa54814f"),
    '5': ObjectId("631c20b37649348dfa548150"),
    '6': ObjectId("631c20b37649348dfa548151")
  }
}
```



{ field: 'value' }

Displaying documents 1 - 7 of 7

```
{
  "_id": ObjectId("631c20b37649348dfa54814b"),
  "name": "Full Cream Milk",
  "types": Array
    0: "Dairy"
    1: "Milk"
  "purchase_date": Object
    day: 28
    month: 9
    year: 2022
  "expiry_date": Object
    day: 2
    month: 9
    year: 2022
  "location": "fridge"
  "amount": "700 ml"
}
```

```
{
  "_id": ObjectId("631c20b37649348dfa54814c"),
  "name": "Chunk Tuna With Chilli",
  "types": Array
    0: "Canned"
    1: "Seafood"
    2: "Fish"
  "purchase_date": Object
    day: 14
    month: 7
    year: 2022
  "expiry_date": Object
    day: 10
    month: 1
    year: 2024
  "location": "pantry"
  "amount": "180 g"
}
```

```
{
  "_id": ObjectId("631c20b37649348dfa54814d"),
  "name": "Broccoli",
  "types": Array
  "purchase_date": Object
}
```

2. I bought 200g of salmon and stored half of it in the fridge and the rest in the freezer.

```
db.foodInventory.insertMany([
  {name: "Salmon", types: ["Seafood", "Fish"],
  purchase_date: {day: 2, month: 9, year: 2022}, expiry_date: {day: 1, month: 11, year: 2022}, location: "freezer", amount: "100 g"},
  {name: "Salmon", types: ["Seafood", "Fish"],
  purchase_date: {day: 2, month: 9, year: 2022}, expiry_date: {day: 4, month: 9, year: 2022}, location: "fridge", amount: "100 g"}])
```

## Inventory.foodInventory

7 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

OPTIONS

FIND

RESET

...

ADD DATA

VIEW

...

...

...

Displaying documents 1 - 7 of 7

REFRESH

```
{
  "_id": ObjectId("631c20b37649348dfa54814b"),
  "name": "Full Cream Milk",
  "types": Array
    0: "Dairy"
    1: "Milk"
  "purchase_date": Object
    day: 25
    month: 9
    year: 2022
  "expiry_date": Object
    day: 2
    month: 9
    year: 2022
  "location": "fridge"
  "amount": "700 ml"
}
```

```
{
  "_id": ObjectId("631c20b37649348dfa54814c"),
  "name": "Chunk Tuna With Chilli",
  "types": Array
    0: "Canned"
    1: "Seafood"
    2: "Fish"
  "purchase_date": Object
    day: 14
    month: 7
    year: 2022
  "expiry_date": Object
    day: 10
    month: 1
    year: 2024
  "location": "pantry"
  "amount": "180 g"
}
```

```
{
  "_id": ObjectId("631c20b37649348dfa54814d"),
  "name": "Broccoli",
  "types": Array
  "purchase_date": Object
}
```

```
Atlas atlas-dr77lw-shard-0 [primary] Inventory> db.foodInventory.insertMany([
  {name: "Salmon", types: ["Seafood", "Fish"],
    purchase_date: {day: 2, month: 9, year: 2022}, expiry_date: {day: 1, month: 11, year: 2022}, location: "freezer", amount: "100 g"},
  {name: "Salmon", types: ["Seafood", "Fish"], purchase_date: {day: 2, month: 9, year: 2022}, expiry_date: {day: 4, month: 9, year: 2022}, location: "fridge", amount: "100 g"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("631c248d7649348dfa548152"),
    '1': ObjectId("631c248d7649348dfa548153")
  }
}
Atlas atlas-dr77lw-shard-0 [primary] Inventory>
```

Documents Aggregations Schema Explain Plan Indexes Validation

① FILTER ( field: 'value' )

➤ OPTIONS FIND RESET ↺ ⋮

➤ ADD DATA ⬇ VIEW [ ] [ ] [ ]

Displaying documents 1 - 9 of 9 < > C REFRESH

```
name: "Oreo"
> types: Array
> purchase_date: Object
> expiry_date: Object
location: "pantry"
amount: "12"

_id: ObjectId('631c248d7649348dfa540152')
name: "Salmon"
types: Array
  0: "Seafood"
  1: "Fish"
purchase_date: Object
  day: 2
  month: 9
  year: 2022
expiry_date: Object
  day: 1
  month: 11
  year: 2022
location: "freezer"
amount: "100 g"

_id: ObjectId('631c248d7649348dfa540153')
name: "Salmon"
types: Array
  0: "Seafood"
  1: "Fish"
purchase_date: Object
  day: 2
  month: 9
  year: 2022
expiry_date: Object
  day: 4
  month: 9
  year: 2022
location: "fridge"
amount: "100 g"
```

3. I bought a 'Sour Cream & Onion Pringles' and stored it in the pantry.

```
db.foodInventory.insertOne({name: "Sour Cream & Onion Pringles", types: ["Snack", "Chips"], purchase_date: {day: 2, month: 9, year: 2022}, expiry_date: {day: 2, month: 7, year: 2023}, location: "pantry", amount: "130 g"})
```

Documents Aggregations Schema Explain Plan Indexes Validation

① FILTER ( field: 'value' )

➤ OPTIONS FIND RESET ↺ ⋮

➤ ADD DATA ⬇ VIEW [ ] [ ] [ ]

Displaying documents 1 - 9 of 9 < > C REFRESH

```
name: "Oreo"
> types: Array
> purchase_date: Object
> expiry_date: Object
location: "pantry"
amount: "12"

_id: ObjectId('631c248d7649348dfa540152')
name: "Salmon"
types: Array
  0: "Seafood"
  1: "Fish"
purchase_date: Object
  day: 2
  month: 9
  year: 2022
expiry_date: Object
  day: 1
  month: 11
  year: 2022
location: "freezer"
amount: "100 g"

_id: ObjectId('631c248d7649348dfa540153')
name: "Salmon"
types: Array
  0: "Seafood"
  1: "Fish"
purchase_date: Object
  day: 2
  month: 9
  year: 2022
expiry_date: Object
  day: 4
  month: 9
  year: 2022
location: "fridge"
amount: "100 g"
```

```

atlas atlas-dr77lw-shard-0 [primary] Inventory> db.foodInventory.insertOne({name: "Sour Cream & Onion Pringles", types:
"Snack", "Chips", purchase_date: {day: 2, month: 9, year: 2022}, expiry_date: {day: 2, month: 7, year: 2023}, location: "pantry", amount: "130 g"})

acknowledged: true,
insertedId: ObjectId("631c26c87649348dfa548154")

```

Inventory.foodInventory 10 1  
DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

amount: "100 g"

```

{
  "_id": ObjectId("631c249d7649348dfa548153"),
  "name": "Salmon",
  "types": Array,
  "purchase_date": Object,
  "expiry_date": Object,
  "location": "fridge",
  "amount": "100 g"
}

```

```

{
  "_id": ObjectId("631c26c87649348dfa548154"),
  "name": "Sour Cream & Onion Pringles",
  "types": Array
    0: "Snack"
    1: "Chips"
  "purchase_date": Object
    day: 2
    month: 9
    year: 2022
  "expiry_date": Object
    day: 2
    month: 7
    year: 2023
  "location": "pantry",
  "amount": "130 g"
}

```

Displaying documents 1 - 10 of 10

## Read:

1. I am planning to make pancakes and the recipe requires 500 ml of milk so I need to find out if I have sufficient milk.

```
.find({"name": /Milk/, "types": "Milk"},{_id: 0, name: 1, amount: 1})
```

Find milk and projects the name and amount.

```

  _id: ObjectId('631c20b37649348dfa54814b')
  name: "Full Cream Milk"
  > types: Array
  > purchase_date: Object
  > expiry_date: Object
    location: "fridge"
    amount: "700 ml"

```

```

Atlas atlas-dr77lw-shard-0 [primary] Inventory> db.foodInventory.find({"name": /Milk/, "types": "Milk"},{_id: 0, name: 1, amount: 1})
[ { name: 'Full Cream Milk', amount: '700 ml' } ]

```

FILTER

`{"name": "/Milk/", "types": "Milk"}`

▼ OPTIONS

PROJECT

`{_id: 0, name: 1, amount: 1}`

SORT

`{ field: -1 } or [['field', -1]]`

MAX TIME MS

60000

COLLATION

`{ locale: 'simple' }`

SKIP

0

LIMIT

0

VIEW

Displaying documents 1 - 1 of 1

<

>

REFRESH

```
name: "Full Cream Milk"
amount: "700 ml"
```

2. I want to view all the expired products.

```
db.foodInventory.find({$or:[{$and: [{"expiry_date.month": {$lt: 9}}, {"expiry_date.year": 2022}], {$and: [{"expiry_date.day": {$lt: 3}}, {"expiry_date.month": 9}, {"expiry_date.year": 2022}], {"expiry_date.year": {$lt: 2022}}]}, {_id:0, name:1, location: 1, amount: 1})
```

Show the name, location, and amount for any products with expiration date before the current date.

```
_id: ObjectId('631c20b37649348dfa54814b')
name: "Full Cream Milk"
> types: Array
> purchase_date: Object
√ expiry_date: Object
  day: 2
  month: 9
  year: 2022
location: "fridge"
amount: "700 ml"
```

---

```
_id: ObjectId('631c20b37649348dfa54814f')
name: "Thickened Cream"
> types: Array
> purchase_date: Object
√ expiry_date: Object
  day: 28
  month: 8
  year: 2022
location: "fridge"
amount: "200 ml"
```

```
Atlas atlas-dr77lw-shard-0 [primary] Inventory> db.foodInventory.find({$or:[{$and: [{"expiry_date.month": {$lt: 9}}, {"expiry_date.year": 2022}]}], {$and: [{"expiry_date.day": {$lt: 3}}, {"expiry_date.month": 9}, {"expiry_date.year": 2022}]}], {"expiry_date.year": {$lt: 2022}}},{_id:0, name:1, location: 1, amount: 1})
[
  { name: 'Full Cream Milk', location: 'fridge', amount: '700 ml' },
  { name: 'Thickened Cream', location: 'fridge', amount: '200 ml' }
]
```

FILTER{"expiry\_date.year": 2022}], {"expiry\_date.year": {\$lt: 2022}}]

OPTIONS

FINDRESET↺⋮

PROJECT{ "\_id":0, name:1, location: 1, amount: 1}

SORT{ field: -1 } or [[ 'field', -1]]MAX TIME MS60000

COLLATION{ locale: 'simple' }SKIP0LIMIT0

VIEW☰{}📄

Displaying documents 1 - 2 of 2<>↻ REFRESH

name: "Full Cream Milk"  
location: "fridge"  
amount: "700 ml"

name: "Thickened Cream"  
location: "fridge"  
amount: "200 ml"

**3. I want to view the amount of meat stored in the freezer.**

```
db.foodInventory.find({"types": "Meat", "location": "freezer"}, {_id:0, name: 1, amount: 1})
```

Finds all documents with the type 'Meat' and storage location as 'freezer'. Prints name and amount.

```

_id: ObjectId('631c20b37649348dfa54814e')
name: "Chicken Breast"
> types: Array
> purchase_date: Object
> expiry_date: Object
  location: "freezer"
  amount: "250 g"

```

```
_id: ObjectId('631c20b37649348dfa548150')
name: "Ground Beef"
> types: Array
> purchase_date: Object
> expiry_date: Object
location: "freezer"
amount: "500 g"
```

```
Atlas atlas-dr77lw-shard-0 [primary] Inventory> db.foodInventory.find({ "types": "Meat", "location": "freezer"}, {_id:0,
name: 1, amount: 1})
[
  { name: 'Chicken Breast', amount: '250 g' },
  { name: 'Ground Beef', amount: '500 g' }
]
Atlas atlas-dr77lw-shard-0 [primary] Inventory>
```

Inventory

foodInventory

10

DOCUMENTS

1

INDEXES

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

1 FILTER

{ "types": "Meat", "location": "freezer" }

▼ OPTIONS

1 PROJECT

{ \_id:0, name: 1, amount: 1 }

1 SORT

{ field: -1 } or [['field', -1]]

1 MAX TIME MS

60000

1 COLLATION

{ locale: 'simple' }

1 SKIP

0

1 LIMIT

0

FIND

RESET

↺

⋮

VIEW

⋮

{ }

⌂

Displaying documents 1 - 2 of 2

<

>

↺ REFRESH

name: "Chicken Breast"

amount: "250 g"

name: "Ground Beef"

amount: "500 g"

## Update:

1. I used 500 ml of the full cream milk to make pancakes so I need to update the amount of milk left.

```
db.foodInventory.updateOne({"name": "Full Cream Milk", "amount" : "700 ml"},{$set:{"amount":"200 ml"}})
```

Finds one document with the name 'Full Cream Milk' and with storage amount '700 ml', and sets the amount in storage to '200 ml'.



```

    _id: ObjectId('631c20b37649348dfa54814b')
    name: "Full Cream Milk"
  > types: Array
  > purchase_date: Object
  > expiry_date: Object
    location: "fridge"
    amount: "700 ml"

```

```

Atlas atlas-dr77lw-shard-0 [primary] Inventory> db.foodInventory.updateOne({"name": "Full Cream Milk", "amount": "700 ml"}, {$set: {"amount": "200 ml"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

```

    _id: ObjectId('631c20b37649348dfa54814b')
    name: "Full Cream Milk"
  > types: Array
  > purchase_date: Object
  > expiry_date: Object
    location: "fridge"
    amount: "200 ml"

```

2. I placed the chicken in the fridge to defrost it so I need to change the location and expiry date.

```

db.foodInventory.updateOne({"name": "Chicken Breast", "location": "freezer" },
{$set: {"location": "fridge", "expiry_date.day": 5, "expiry_date.month": 9,
"expiry_date.year": 2022}})

```

Finds one document with the name 'Chicken Breast', and the storage's location be 'freezer'. Sets the location in storage to 'fridge', and the expiry date to an earlier date.

```

    _id: ObjectId('631c20b37649348dfa54814e')
    name: "Chicken Breast"
  > types: Array
  > purchase_date: Object
  > expiry_date: Object
    day: 1
    month: 6
    year: 2023
    location: "freezer"
    amount: "250 g"

```

```

Atlas atlas-dr77lw-shard-0 [primary] Inventory> db.foodInventory.updateOne({"name": "Chicken Breast", "location": "freezer" }, {$set: {"location": "fridge", "expiry_date.day": 5, "expiry_date.month": 9, "expiry_date.year": 2022}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

```

    _id: ObjectId('631c20b37649348dfa54814e')
    name: "Chicken Breast"
  > types: Array
  > purchase_date: Object
  > expiry_date: Object
    day: 5
    month: 9
    year: 2022
  location: "fridge"
  amount: "250 g"

```

3. I ate all the Oreo cookies in the pantry..

```
db.foodInventory.updateOne({"name":/Oreo/, "location": "pantry"},{$set:{amount:"0"}})
```

Finds one document with the name 'Oreo' and 'pantry' as storage location. Set the amount to 0

```

    _id: ObjectId('631c20b37649348dfa548151')
    name: "Oreo"
  > types: Array
  > purchase_date: Object
  > expiry_date: Object
  location: "pantry"
  amount: "12"

```

```

Atlas atlas-dr77lw-shard-0 [primary] Inventory> db.foodInventory.updateOne({ name:/Oreo/, location: "pantry"},{$set:{
amount:"0"}})

acknowledged: true,
insertedId: null,
matchedCount: 1,
modifiedCount: 1,
upsertedCount: 0

```

```

    _id: ObjectId('631c20b37649348dfa548151')
    name: "Oreo"
  > types: Array
  > purchase_date: Object
  > expiry_date: Object
  location: "pantry"
  amount: "0"

```

## Delete:

1. I want to remove products that have already been consumed from the database

```
db.foodInventory.deleteMany({amount: "0"})
```

Delete all documents that have '0' for amount.

The screenshot shows the MongoDB Atlas web interface for the `inventory.foodInventory` collection. At the top right, it displays **10** DOCUMENTS and **1** INDEXES. The navigation bar includes tabs for Documents, Aggregations, Schema, Explain Plan, Indexes, and Validation. The `Documents` tab is active, showing a filter bar with the query `{amount: "0"}` and buttons for `OPTIONS`, `FIND`, `RESET`, and a refresh icon. Below the filter bar, there's a section for `ADD DATA` and a `VIEW` section with icons for list, JSON, and table views. The main area displays a single document in JSON format: 

```
{_id: ObjectId('631c20b37649348dfa548151'), name: "Oreo", types: Array, purchase_date: Object, expiry_date: Object, location: "pantry", amount: "0"}
```

. At the bottom, a terminal window shows the execution of the `deleteMany` command: 

```
tlas atlas-dr77lw-shard-0 [primary] Inventory> db.foodInventory.deleteMany({amount: "0"})
acknowledged: true, deletedCount: 1
tlas atlas-dr77lw-shard-0 [primary] Inventory>
```

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER {amount: "0"}

OPTIONS

FIND

RESET



ADD DATA



VIEW



Displaying documents 0 - 0 of 0



REFRESH



## No results

Try to modify your query to get results