



THE TECHNOLOGICAL UNIVERSITY OF THE
SHANNON: MIDLANDS MIDWEST

Student Name: **Parimal Ganesh Sawant**

Student Id Number: **A00325752**

Course: **Masters in Data Analytics**

Title of Assignment: **Advanced Databases Project**

Date: **01 May 2025**

Title: **College Scholarship Sponsors**

Declaration

I hereby certify that the material, which is submitted in this report towards the award of MSc. in Data Analytics, is entirely my own work and has not been submitted for any academic assessment other than part fulfilment of the above named award.

Signed: Parimal Ganesh Sawant

Date: 01/05/2025

Table of Contents

1. Feature Checklist	Page 4
2. Section 1: Project Overview	Page 5
3. Section 2: Mongo Shell Simple Queries	Page 8
4. Section 3: Mongo Shell – Aggregation Framework	Page 42
5. Section 4: Atlas Overview and Simple Queries	Page 54
6. Section 5: Atlas Aggregation Framework	Page 60
7. Section 6: Atlas Dashboard / Issues Faced with Atlas	Page 63
8. Section 7: Compass Overview and Simple Queries	Page 65
9. Section 8: Replica set	Page 68
10. Section 9: Appendices	Page 71

Feature	Simple Query	Page No.
\$all to match multiple array elements	Q1	Page 8
\$gte with nested fields	Q1, Q3, Q5, Q10, Q12	Pages 8, 12, 16, 26, 30
Dot notation for nested fields	Q1–Q17	Pages 8 - 40
\$gt for filtering numeric fields	Q2, Q8, Q9, Q14	Pages 10, 22, 24, 34
\$in to match multiple values	Q3, Q4, Q17	Pages 12, 14, 40
\$or and \$min combination	Q11	Page 28
\$nor for excluding conditions	Q5	Page 16
Array index-based querying (.0, .2)	Q6, Q12	Pages 18, 30
\$elemMatch with regex	Q7	Page 20
\$size to match array length	Q7, Q11, Q13, Q16	Pages 20, 28, 32, 38
\$ne (not equal)	Q8, Q14	Pages 22, 34
Sorting with .sort()	Q9, Q17	Page 24, 40
Combining \$in and \$not inside array	Q10, Q16	Pages 8–9
\$nin to exclude specific values	Q11, Q17	Pages 28, 40
Feature	Aggregation Query	Page No.
\$match for filtering data	Agg Q1–Q9	Pages 42 - 52
\$group with \$sum for totals	Agg Q1, Q3, Q4, Q7	Pages 42, 44, 45, 49
\$group with \$avg , \$min , \$max	Agg Q2, Q8	Pages 43, 51
\$group with \$first , \$avg	Agg Q9	Page 52
\$sort ascending/descending	Agg Q1, Q2, Q4, Q7, Q8	Pages 42, 43, 45, 49, 51
\$project with \$cond for categorisation	Agg Q3, Q8	Page 44, 51
\$addFields for computed fields	Agg Q4, Q7, Q8	Pages 45, 49
\$unwind to flatten arrays	Agg Q2, Q5, Q7, Q9	Pages 43, 46, 49, 51
\$facet for parallel pipelines	Agg Q5	Page 46
\$bucket for fixed-range bucketing	Agg Q6	Page 47
\$bucketAuto for dynamic-range bucketing	Agg Q6 (2nd part)	Page 48
\$out to write aggregation results to collection	Agg Q6	Page 46
\$addToSet for counting unique values	Agg Q7, Q8	Pages 49, 51
\$round for formatting numerical results	Agg Q8	Page 51

Section 1: Project Overview

The project involves setting up a MongoDB database to import and store the data into database which contains 1000 scholarship records. Below is the procedure for setting up a database.

1. **Verifying MongoDB installation:** The command mongod --version, confirms that MongoDB is installed.



```
Command Prompt
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\parim>mongod --version
db version v8.0.6
Build Info: {
    "version": "8.0.6",
    "gitVersion": "80f21521ad4a3dfd5613f5d649d7058c6d46277f",
    "modules": [],
    "allocator": "tcmalloc-gperf",
    "environment": {
        "distmod": "windows",
        "distarch": "x86_64",
        "target_arch": "x86_64"
    }
}
C:\Users\parim>
```

Section 1.1

2. **Starting MongoDB on a port:** If you have configured the replica set then it is essential to start up with the port and initialise the replica set. Which in my case the replica set was already configured so it is showing it as [direct: secondary].



```
mongosh mongodb://127.0.0.1:27017
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

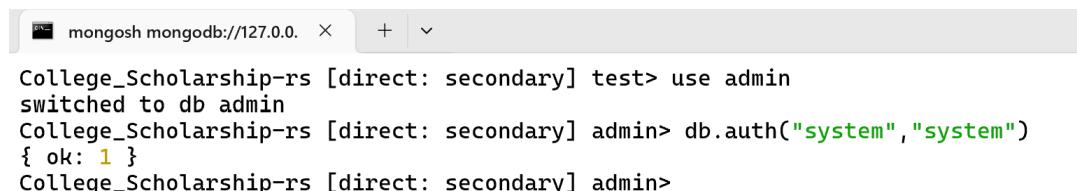
C:\Users\parim>mongod --version
db version v8.0.6
Build Info: {
    "version": "8.0.6",
    "gitVersion": "80f21521ad4a3dfd5613f5d649d7058c6d46277f",
    "modules": [],
    "allocator": "tcmalloc-gperf",
    "environment": {
        "distmod": "windows",
        "distarch": "x86_64",
        "target_arch": "x86_64"
    }
}
C:\Users\parim>mongosh --port 27017
Current Mongosh Log ID: 68117d7dd2a5c3b73eb5f898
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.0
Using MongoDB: 8.0.6
Using Mongosh: 2.5.0

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
College_Scholarship-rs [direct: secondary] test> |
```

Section 1.2

3. **Create a Database User:** Connecting to the admin user to create a new user. Once it is connected to admin a new user needs to be created, which in my database was already created as system and in admin the new user must be authenticated by username and password which is db.auth("system", "system"). To create a new user below command can be used:

```
db.createUser( {
    user: "system",
    pwd: "system",
    roles: ["root"]
})
```



```
mongosh mongodb://127.0.0.1:27017
College_Scholarship-rs [direct: secondary] test> use admin
switched to db admin
College_Scholarship-rs [direct: secondary] admin> db.auth("system", "system")
{ ok: 1 }
College_Scholarship-rs [direct: secondary] admin>
```

Section 1.3

- 4. Creating the database:** Before importing the json file a database should be created which is mentioned in the mongoimport command.

```
mongosh mongodb://127.0.0.1
College_Scholarship-rs [direct: secondary] admin> use College_Scholarship_Details
switched to db College_Scholarship_Details
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> |
```

Section 1.4

- 5. Importing the json file:** Since I have connected the database with replica set and with authenticator it was necessary to specify the username, password, all the localhost ports of replicaset and the replicaset name to have the perfect connection.

```
Command Prompt
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\parim>mongoimport "mongodb://system:system@localhost:27017,localhost:27018,localhost:27019/?replicaSet=College_Scholarship-rs&authSource=admin" --collection Scholarship --file "C:\Mongo_Project\Dataset\College_Scholarship_Sponsor.json" --jsonArray --drop
2025-04-30T02:51:31.320+0100      connected to: mongodb://[**REDACTED**]@localhost:27017,localhost:27018,localhost:27019/?replicaSet=College_Scholarsh
ip-rsAuthSource=admin
2025-04-30T02:51:31.320+0100      dropping: test.Scholarship
2025-04-30T02:51:31.383+0100      1000 document(s) imported successfully. 0 document(s) failed to import.

C:\Users\parim>
```

Section 1.5

- 6. Verifying the import:** To verify the import use db.Scholarship.countDocuments() this will return the total number of documents inside the collection.

```
mongosh mongodb://127.0.0.1
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.countDocuments()
1000
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>
```

Section 1.6

- 7. Querying a single document:** Displaying a simple document to check its structure using db.Scholarship.find().pretty()

```
mongosh mongodb://127.0.0.1
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.find().pretty()
[
  {
    "_id": 8,
    "Title": "Code4All",
    "Discipline": [ "Medicine", "Arts" ],
    "Breakdown": [
      "Free Fees",
      "Free Campus Accommodation",
      "Free Books",
      "Laptop Grant"
    ],
    "Annual_Value": 10000,
    "Duration": 3,
    "Initial_Payment_Num": 345,
    "Num_Instalments": 87,
    "Category": "Postgraduate",
    "Conditions": "Must pass exams",
    "College": { "Name": "DIT", "City": "Dublin", "Num_Students": 3550 },
    "Sponsor": {
      "Name": "Aer Lingus",
      "Sector": "Airlines",
      "Headquarters": "Ireland",
      "Num_Staff": 25920
    }
  }
]
```

Section 1.7

Array Fields

Discipline Array: List of academic disciplines eligible for scholarships.

- **Values:** Arts, Business, Engineering, Medicine, Science

```
[mongosh mongoDB://127.0.0.1:27017] + v
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.distinct("Discipline")
[ 'Arts', 'Business', 'Engineering', 'Medicine', 'Science' ]
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> |
```

Section 1.8

Breakdown Array: Consist list of benefits provided by scholarships.

- **Values:** Additional Tutor Support, Free Books, Free Campus Accommodation, Free Fees, Laptop Grant

```
[mongosh mongoDB://127.0.0.1:27017] + v
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.distinct("Breakdown")
[
  'Additional Tutor Support',
  'Free Books',
  'Free Campus Accommodation',
  'Free Fees',
  'Laptop Grant'
]
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> |
```

Section 1.9

Embedded Document

Document 1: College

- **Subfields:** Name, City, Num_Students

Document 2: Sponsor

- **Subfields:** Name, Sector, Headquarters, Num_Staff

Section 2: Mongo Shell Simple Queries

Query 1

Question: Display the Scholarship Title, College Name, Sponsor Name of postgraduate scholarships students in **Dublin** offer **free fees** and **laptop grant** as breakdown, that are also sponsored by companies in the **Technology** sector, and have at least annual value of **€14000**.

Mongo Db Count command

```
db.Scholarship.countDocuments ( 
    {
        Category: "Postgraduate",
        "College.City": "Dublin",
        Breakdown: { $all: ["Free Fees", "Laptop Grant"] },
        "Sponsor.Sector": "Technology",
        iAnnual_Value: { $gte: 14000 }
    }
)
mongosh mongodb://127.0.0.1:27017/College_Scholarship_rs> + <
College_Scholarship_rs [direct: secondary] College_Scholarship_Details> db.Scholarship.countDocuments ( 
... {
...     Category: "Postgraduate",
...     "College.City": "Dublin",
...     Breakdown: { $all: ["Free Fees", "Laptop Grant"] },
...     "Sponsor.Sector": "Technology",
...     iAnnual_Value: { $gte: 14000 }
... }
...
5
College_Scholarship_rs [direct: secondary] College_Scholarship_Details> |
```

Image 1.1.1

Mongo Db Find command

```
db.Scholarship.find ( 
    {
        Category: "Postgraduate",
        "College.City": "Dublin",
        Breakdown: { $all: ["Free Fees", "Laptop Grant"] },
        "Sponsor.Sector": "Technology",
        iAnnual_Value: { $gte: 14000 }
    },
    {
        Title: 1,
        Category: 1,
        iAnnual_Value: 1,
        Breakdown: 1,
        "Sponsor.Name": 1,
        "Sponsor.Sector": 1,
        "College.City": 1,
        "College.Name": 1
    }
)
```

```

mongosh mongodb://127.0.0.1:27017/College_Scholarship_Details> db.Scholarship.find(
... {
...   Category: "Postgraduate",
...   "College.City": "Dublin",
...   Breakdown: { $all: ["Free Fees", "Laptop Grant"] },
...   "Sponsor.Sector": "Technology",
...   iAnnual_Value: { $gte: 14000 }
... },
... {
...   Title: 1,
...   Category: 1,
...   iAnnual_Value: 1,
...   Breakdown: 1,
...   "Sponsor.Name": 1,
...   "Sponsor.Sector": 1,
...   "College.City": 1,
...   "College.Name": 1
... }
... )
...
[
{
  _id: 392,
  Title: 'InspireU',
  Breakdown: [ 'Free Fees', 'Free Campus Accommodation', 'Laptop Grant' ],
  iAnnual_Value: 15000,
  Category: 'Postgraduate',
  College: { Name: 'DCU', City: 'Dublin' },
  Sponsor: { Name: 'Intel', Sector: 'Technology' }
},
{
  _id: 24,
  Title: 'HopeSchol',
  Breakdown: [ 'Free Fees', 'Free Campus Accommodation', 'Laptop Grant' ],
  iAnnual_Value: 15000,
  Category: 'Postgraduate',
  College: { Name: 'Royal College Of Surgeons ', City: 'Dublin' },
  Sponsor: { Name: 'Google', Sector: 'Technology' }
],

```

Image 1.2.1

```

mongosh mongodb://127.0.0.1:27017/College_Scholarship_Details>
...
{
  _id: 24,
  Title: 'HopeSchol',
  Breakdown: [ 'Free Fees', 'Free Campus Accommodation', 'Laptop Grant' ],
  iAnnual_Value: 15000,
  Category: 'Postgraduate',
  College: { Name: 'Royal College Of Surgeons ', City: 'Dublin' },
  Sponsor: { Name: 'Google', Sector: 'Technology' }
},
{
  _id: 885,
  Title: 'InspireU',
  Breakdown: [
    'Free Books',
    'Additional Tutor Support',
    'Laptop Grant',
    'Free Fees'
  ],
  iAnnual_Value: 15000,
  Category: 'Postgraduate',
  College: { Name: 'National College Of Ireland', City: 'Dublin' },
  Sponsor: { Name: 'Intel', Sector: 'Technology' }
},
{
  _id: 902,
  Title: 'InspireU',
  Breakdown: [ 'Free Fees', 'Laptop Grant' ],
  iAnnual_Value: 14000,
  Category: 'Postgraduate',
  College: { Name: 'Colaiste Mhuire Marino', City: 'Dublin' },
  Sponsor: { Name: 'Intel', Sector: 'Technology' }
},
{
  _id: 639,
  Title: 'InspireU',
  Breakdown: [
    'Additional Tutor Support',
    'Laptop Grant',
    'Free Fees',
    'Free Campus Accommodation'
  ],
  iAnnual_Value: 15000,
  Category: 'Postgraduate',
  College: { Name: 'American College', City: 'Dublin' },
  Sponsor: { Name: 'Intel', Sector: 'Technology' }
}
]
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>

```

Image 1.2.2

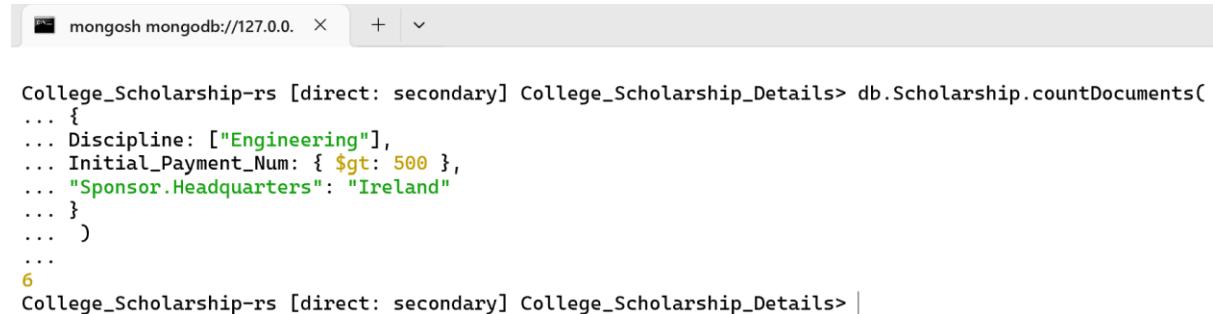
In this query we are using \$all to match the documents in the database with both “Free Fees” and “Laptop Grant” in the Breakdown array and \$gte to filter out the values greater then equal to €14000. We have also used dot notation to make use of nested fields for Postgraduate scholarships in Dublin sponsored by Technology sector companies.

Query 2

Question: Show the title, sponsor name and college name for scholarship with discipline in **Engineering**, initial payment above **€500** and sponsored by companies in **Ireland**.

Mongo Db Count command

```
db.Scholarship.countDocuments(  
  {  
    Discipline: ["Engineering"],  
    Initial_Payment_Num: { $gt: 500 },  
    "Sponsor.Headquarters": "Ireland"  
  }  
)
```



A screenshot of a terminal window titled 'mongosh mongodb://127.0.0.1:27017'. The command entered is 'db.Scholarship.countDocuments({ Discipline: ["Engineering"], Initial_Payment_Num: { \$gt: 500 }, "Sponsor.Headquarters": "Ireland" })'. The output shows the number '6' indicating the count of documents matching the criteria.

```
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.countDocuments(  
... {  
...   Discipline: ["Engineering"],  
...   Initial_Payment_Num: { $gt: 500 },  
...   "Sponsor.Headquarters": "Ireland"  
... }  
... )  
...  
6  
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> |
```

Image 2.1.1

Mongo Db Find command

```
db.Scholarship.find(  
  {  
    Discipline: ["Engineering"],  
    Initial_Payment_Num: { $gt: 500 },  
    "Sponsor.Headquarters": "Ireland"  
  },  
  {  
    Title: 1,  
    Initial_Payment_Num: 1,  
    Discipline: 1,  
    "College.Name": 1,  
    "Sponsor.Name": 1,  
    "Sponsor.Headquarters": 1,  
    _id: 0  
  }  
)
```

```

mongosh mongoDB://127.0.0.1 + v
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.find(
... {
...     Discipline: ["Engineering"],
...     Initial_Payment_Num: { $gt: 500 },
...     "Sponsor.Headquarters": "Ireland"
... },
... {
...     Title: 1,
...     Initial_Payment_Num: 1,
...     Discipline: 1,
...     "College.Name": 1,
...     "Sponsor.Name": 1,
...     "Sponsor.Headquarters": 1,
...     _id: 0
... }
... )
...
[
{
    Title: 'Code4All',
    Discipline: [ 'Engineering' ],
    Initial_Payment_Num: 563,
    College: { Name: 'Athlone IT' },
    Sponsor: { Name: 'Aer Lingus', Headquarters: 'Ireland' }
},
{
    Title: 'GradFund',
    Discipline: [ 'Engineering' ],
    Initial_Payment_Num: 608,
    College: { Name: 'College' },
    Sponsor: { Name: 'Irish Life', Headquarters: 'Ireland' }
},
{
    Title: 'NextGen Grant',
    Discipline: [ 'Engineering' ],
    Initial_Payment_Num: 836,
    College: { Name: 'Setanta College' },
    Sponsor: { Name: 'DCC plc', Headquarters: 'Ireland' }
},
{
    Title: 'NextGen Grant',
    Discipline: [ 'Engineering' ],
    Initial_Payment_Num: 667,
    College: { Name: 'Jordanstown' },
    Sponsor: { Name: 'DCC plc', Headquarters: 'Ireland' }
},
{
    Title: 'GradFund',
    Discipline: [ 'Engineering' ],
    Initial_Payment_Num: 896,
    College: { Name: 'Griffith College Dublin' },
    Sponsor: { Name: 'Irish Life', Headquarters: 'Ireland' }
},
{
    Title: 'GreenEdu',
    Discipline: [ 'Engineering' ],
    Initial_Payment_Num: 836,
    College: { Name: 'DIT' },
    Sponsor: { Name: 'CRH plc', Headquarters: 'Ireland' }
}
]
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>

```

Image 2.2.1

```

{
    Title: 'NextGen Grant',
    Discipline: [ 'Engineering' ],
    Initial_Payment_Num: 667,
    College: { Name: 'Jordanstown' },
    Sponsor: { Name: 'DCC plc', Headquarters: 'Ireland' }
},
{
    Title: 'GradFund',
    Discipline: [ 'Engineering' ],
    Initial_Payment_Num: 896,
    College: { Name: 'Griffith College Dublin' },
    Sponsor: { Name: 'Irish Life', Headquarters: 'Ireland' }
},
{
    Title: 'GreenEdu',
    Discipline: [ 'Engineering' ],
    Initial_Payment_Num: 836,
    College: { Name: 'DIT' },
    Sponsor: { Name: 'CRH plc', Headquarters: 'Ireland' }
}
]
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>

```

Image 2.2.2

The above query is a simple one with only 1 criteria of \$gt which is filtering out documents with initial payments greater than €500. We have also used array matching to select scholarships from Engineering Discipline and dot notation to ensure that the headquarters are in Ireland.

Query 3

Question: Show id, Title, Discipline, College Name of all the **postgraduate** students which are located in city **Galway, Galway / Castlebar** with **free accommodation**, valued atleast **€11000**.

Mongo Db Count command

```
db.Scholarship.countDocuments(  
  {  
    Category: "Postgraduate",  
    "College.City": { $in: ["Galway", "Galway / Castlebar"] },  
    Breakdown: "Free Campus Accommodation",  
    iAnnual_Value: { $gte: 11000 }  
  }  
)  
mongosh mongodb://127.0.0.1:27017  
+  
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.countDocuments(  
... {  
...   Category: "Postgraduate",  
...   "College.City": { $in: ["Galway", "Galway / Castlebar"] },  
...   Breakdown: "Free Campus Accommodation",  
...   iAnnual_Value: { $gte: 11000 }  
... }  
... )  
4
```

Image 3.1.1

Mongo Db Find command

```
db.Scholarship.find(  
  {  
    Category: "Postgraduate",  
    "College.City": { $in: ["Galway", "Galway / Castlebar"] },  
    Breakdown: "Free Campus Accommodation",  
    iAnnual_Value: { $gte: 11000 }  
  },  
  {  
    Title: 1,  
    Discipline: 1,  
    iAnnual_Value: 1,  
    "College.Name": 1,  
    "College.City": 1,  
    Breakdown: 1,  
    _id: 0  
  })
```

```

College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.find(
... {
...   Category: "Postgraduate",
...   "College.City": { $in: ["Galway", "Galway / Castlebar"] },
...   Breakdown: "Free Campus Accommodation",
...   iAnnual_Value: { $gte: 11000 }
... },
... {
...   Title: 1,
...   Discipline: 1,
...   iAnnual_Value: 1,
...   "College.Name": 1,
...   "College.City": 1,
...   Breakdown: 1,
...   _id: 0
... }
...
[
{
  Title: 'SmartStep',
  Discipline: [ 'Science' ],
  Breakdown: [
    'Free Fees',
    'Free Campus Accommodation',
    'Free Books',
    'Laptop Grant'
  ],
  iAnnual_Value: 11000,
  College: { Name: 'GMIT', City: 'Galway / Castlebar' }
},
{
  Title: 'SkillRise',
  Discipline: [ 'Business' ],
  Breakdown: [
    'Free Campus Accommodation',
    'Free Books',
    'Additional Tutor Support',
    'Laptop Grant'
  ],
  iAnnual_Value: 14000,
  College: { Name: 'NUIG', City: 'Galway' }
},
{
  Title: 'GradFund',
  Discipline: [ 'Engineering', 'Medicine', 'Business', 'Arts' ],
  Breakdown: [ 'Free Fees', 'Free Campus Accommodation', 'Laptop Grant' ],
  iAnnual_Value: 13000,
  College: { Name: 'NUIG', City: 'Galway' }
},
{
  Title: 'SkillRise',
  Discipline: [ 'Business', 'Arts', 'Science', 'Engineering' ],
  Breakdown: [
    'Free Books',
    'Additional Tutor Support',
    'Laptop Grant',
    'Free Campus Accommodation'
  ],
  iAnnual_Value: 12000,
  College: { Name: 'NUIG', City: 'Galway' }
}
]

```

Image 3.2.1

```

  iAnnual_Value: 14000,
  College: { Name: 'NUIG', City: 'Galway' }
},
{
  Title: 'GradFund',
  Discipline: [ 'Engineering', 'Medicine', 'Business', 'Arts' ],
  Breakdown: [ 'Free Fees', 'Free Campus Accommodation', 'Laptop Grant' ],
  iAnnual_Value: 13000,
  College: { Name: 'NUIG', City: 'Galway' }
},
{
  Title: 'SkillRise',
  Discipline: [ 'Business', 'Arts', 'Science', 'Engineering' ],
  Breakdown: [
    'Free Books',
    'Additional Tutor Support',
    'Laptop Grant',
    'Free Campus Accommodation'
  ],
  iAnnual_Value: 12000,
  College: { Name: 'NUIG', City: 'Galway' }
}
]
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>

```

Image 3.2.2

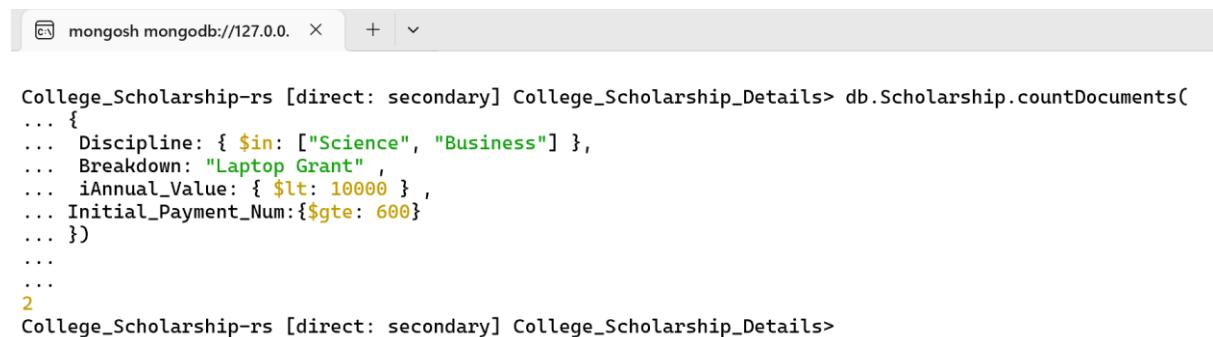
In this query we have made use of \$in to match scholarship in Galway or Galway / Castlebar and string matching for Free Campus Accommodation, we also applied \$gte to filter annual value greater than equal to €10000.

Query 4

Question: Show id, Title, Discipline of those scholarships where discipline is **Science** or **Business**, must include **Laptop Grant** in Breakdown, has an annual value less then or equal to **€10000**, and has an initial payment number atleast **€600**.

Mongo Db Count command

```
db.Scholarship.countDocuments(  
  {  
    Discipline: { $in: ["Science", "Business"] },  
    Breakdown: "Laptop Grant",  
    iAnnual_Value: { $lt: 10000 } ,  
    Initial_Payment_Num:{$gte: 600}  
})
```



The screenshot shows a terminal window titled 'mongosh mongodb://127.0.0.' with a single tab. The command entered is a MongoDB query to count documents in the 'Scholarship' collection. The query filters for scholarships where the Discipline is either 'Science' or 'Business', the Breakdown is 'Laptop Grant', the annual value is less than 10000, and the initial payment number is at least 600. The output shows the command and its result, which is the count of matching documents.

```
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.countDocuments(  
... {  
...   Discipline: { $in: ["Science", "Business"] },  
...   Breakdown: "Laptop Grant" ,  
...   iAnnual_Value: { $lt: 10000 } ,  
...   Initial_Payment_Num:{$gte: 600}  
... })  
...  
...  
2  
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>
```

Image 4.1.1

Mongo Db Find command

```
db.Scholarship.find(  
  {  
    Discipline: { $in: ["Science", "Business"] },  
    Breakdown: "Laptop Grant",  
    iAnnual_Value: { $lt: 10000 } ,  
    Initial_Payment_Num:{$gte: 600}  
  },  
  {  
    Title: 1,  
    Discipline: 1,  
    iAnnual_Value: 1,  
    Breakdown: 1,  
    Initial_Payment_Num: 1  
  })
```

```
mongosh mongodb://127.0.0.1:27017/College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.find(
... {
...   Discipline: { $in: ["Science", "Business"] },
...   Breakdown: "Laptop Grant",
...   iAnnual_Value: { $lt: 10000 },
...   Initial_Payment_Num: {$gte: 600}
... },
... {
...   Title: 1,
...   Discipline: 1,
...   iAnnual_Value: 1,
...   Breakdown: 1,
...   Initial_Payment_Num: 1
... }
... )
...
[
  {
    _id: 387,
    Title: 'EduFund',
    Discipline: [ 'Medicine', 'Business' ],
    Breakdown: [ 'Free Fees', 'Laptop Grant' ],
    iAnnual_Value: 9000,
    Initial_Payment_Num: 600
  },
  {
    _id: 388,
    Title: 'HopeSchol',
    Discipline: [ 'Science', 'Engineering', 'Medicine', 'Business' ],
    Breakdown: [
      'Additional Tutor Support',
      'Laptop Grant',
      'Free Fees',
      'Free Campus Accommodation'
    ],
    iAnnual_Value: 9000,
    Initial_Payment_Num: 600
  }
]
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>
```

Image 4.2.1

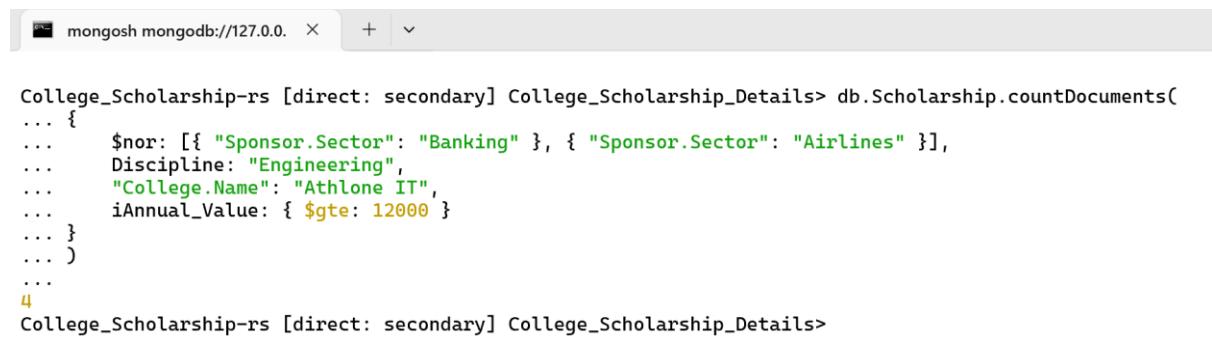
In this query, we have used `$in` to select between Science or Business disciplines and `$lt` for annual values less than €10000 and `$gte` for initial payments greater than equal to 600, retrieving relevant fields.

Query 5

Question: Display the id, title, College Name where Sponsor sector not equal to Banking or Airlines, Discipline consist of Engineering, college name must be Athlone IT and Annual Scholarship Value should be greater then 12000.

Mongo Db Count command

```
db.Scholarship.countDocuments(  
    {  
        $nor: [{ "Sponsor.Sector": "Banking" }, { "Sponsor.Sector": "Airlines" }],  
        Discipline: "Engineering",  
        "College.Name": "Athlone IT",  
        iAnnual_Value: { $gte: 12000 }  
    }  
)
```



```
mongosh mongodb://127.0.0. × + ▾  
  
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.countDocuments(  
... {  
...     $nor: [{ "Sponsor.Sector": "Banking" }, { "Sponsor.Sector": "Airlines" }],  
...     Discipline: "Engineering",  
...     "College.Name": "Athlone IT",  
...     iAnnual_Value: { $gte: 12000 }  
... }  
... }  
4  
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>
```

Image 5.1.1

Mongo Db Find command

```
db.Scholarship.find(  
    {  
        $nor: [{ "Sponsor.Sector": "Banking" }, { "Sponsor.Sector": "Airlines" }],  
        Discipline: "Engineering",  
        "College.Name": "Athlone IT",  
        iAnnual_Value: { $gte: 12000 }  
    },  
    {  
        Title: 1,  
        "College.Name": 1,  
        "Sponsor.Sector": 1,  
        Discipline: 1,  
        iAnnual_Value: 1  
    }  
)
```

```

mongosh mongodb://127.0.0.1:27017/College_Scholarship_rs [direct: secondary] College_Scholarship_Details> db.Scholarship.find(
... {
...   $nor: [{ "Sponsor.Sector": "Banking" }, { "Sponsor.Sector": "Airlines" }],
...   "College.Name": "Athlone IT",
...   iAnnual_Value: { $gte: 12000 },
... },
... {
...   Title: 1,
...   "College.Name": 1,
...   "Sponsor.Sector": 1,
...   Discipline: 1,
...   iAnnual_Value: 1
... }
...
[{
  _id: 40,
  Title: 'EduFund',
  Discipline: [ 'Business', 'Arts', 'Science', 'Engineering' ],
  iAnnual_Value: 12000,
  College: { Name: 'Athlone IT' },
  Sponsor: { Sector: 'Telecommunications' }
},
{
  _id: 63,
  Title: 'SkillRise',
  Discipline: [ 'Engineering', 'Medicine', 'Business', 'Arts' ],
  iAnnual_Value: 15000,
  College: { Name: 'Athlone IT' },
  Sponsor: { Sector: 'Manufacturing' }
},
{
  _id: 959,
  Title: 'BrightPath',
  Discipline: [ 'Science', 'Engineering' ],
  iAnnual_Value: 13000,
  College: { Name: 'Athlone IT' },
  Sponsor: { Sector: 'Manufacturing' }
},
]

```

Image 5.2.1

```

mongosh mongodb://127.0.0.1:27017/College_Scholarship_rs [direct: secondary] College_Scholarship_Details>
},
{
  _id: 959,
  Title: 'BrightPath',
  Discipline: [ 'Science', 'Engineering' ],
  iAnnual_Value: 13000,
  College: { Name: 'Athlone IT' },
  Sponsor: { Sector: 'Manufacturing' }
},
{
  _id: 577,
  Title: 'GreenEdu',
  Discipline: [ 'Science', 'Engineering', 'Arts' ],
  iAnnual_Value: 14000,
  College: { Name: 'Athlone IT' },
  Sponsor: { Sector: 'Construction' }
}
]
```

Image 5.2.2

In this query, we use \$nor to exclude the sponsors from Banking or Airlines sector and string match for Discipline in Athlone IT, the \$gte criteria filters out annual values greater than equal to €12000, selecting specific scholarship field.

Query 6

Question: Show the id, Title where the breakdown features following array position, **Free Fees** on position 1 and **Laptop Grant** on position 3, name of sponsor is **Ryanair**, duration of study is **4** years and total number of installments must be atleast **80**.

Mongo Db Count command

```
db.Scholarship.countDocuments(  
{  
    "Breakdown.0": "Free Fees",  
    "Breakdown.2": "Laptop Grant",  
    "Sponsor.Name": "Ryanair",  
    Duration: {$eq: 4},  
    Num_Instalments: {$gte: 80},  
}  
)  
  
mongosh mongodb://127.0.0.1:27017  
+ v  
  
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.countDocuments(  
... {  
...     "Breakdown.0": "Free Fees",  
...     "Breakdown.2": "Laptop Grant",  
...     "Sponsor.Name": "Ryanair",  
...     Duration: {$eq: 4},  
...     Num_Instalments: {$gte: 80},  
... }  
... }  
...  
2  
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> |
```

Image 6.1.1

Mongo Db Find command

```
db.Scholarship.find(  
{  
    "Breakdown.0": "Free Fees",  
    "Breakdown.2": "Laptop Grant",  
    "Sponsor.Name": "Ryanair",  
    Duration: {$eq: 4},  
    Num_Instalments: {$gte: 80},  
},  
{  
    Title: 1,  
    Breakdown: 1,  
    "Sponsor.Name": 1,  
    Duration: 1,  
    Num_Instalments: 1  
})
```

```
mongosh mongodb://127.0.0.1:27017/College_Scholarship_rs [direct: secondary] College_Scholarship_Details> db.Scholarship.find({  
    ...  
    ...  
    "Breakdown.0": "Free Fees",  
    ...  
    "Breakdown.2": "Laptop Grant",  
    ...  
    "Sponsor.Name": "Ryanair",  
    ...  
    Duration: {$eq: 4},  
    ...  
    Num_Instalments: {$gte: 80},  
    ...  
    ...  
    ...  
    Title: 1,  
    ...  
    ...  
    Breakdown: 1,  
    ...  
    "Sponsor.Name": 1,  
    ...  
    Duration: 1,  
    ...  
    Num_Instalments: 1  
    ...  
    ...  
})  
[  
    {  
        _id: 292,  
        Title: 'SmartStep',  
        Breakdown: [ 'Free Fees', 'Free Books', 'Laptop Grant' ],  
        Duration: 4,  
        Num_Instalments: 82,  
        Sponsor: { Name: 'Ryanair' }  
    },  
    {  
        _id: 602,  
        Title: 'SmartStep',  
        Breakdown: [ 'Free Fees', 'Free Books', 'Laptop Grant' ],  
        Duration: 4,  
        Num_Instalments: 90,  
        Sponsor: { Name: 'Ryanair' }  
    }  
]  
College_Scholarship_rs [direct: secondary] College_Scholarship_Details>
```

Image 6.2.1

In this query, we are using array positional indexing (Breakdown.0 and Breakdown.2) to match Free Fees and Laptop Grant at position 1 and 3 and \$eq for a scholarship equal to 4 years duration. We applied \$gte for scholarships greater then equal to 80 installments and string match for Sponsor Name as Ryanair.

Query 7

Question: Find the title, College Name, Sponsor Name where Breakdown includes the word “Free”, company headquarters in **United Kingdom**, whose college is located in **Dublin**, the size of discipline array is equal to **3**, and Annual sponsorship value is more than **13000**.

Mongo Db Count command

```
db.Scholarship.countDocuments(  
{  
    Breakdown: { $elemMatch: { $regex: /^Free/ } },  
    "Sponsor.Headquarters": "United Kingdom",  
    "College.City": "Dublin",  
    Discipline: { $size: 3 },  
    iAnnual_Value: { $lt: 10000 }  
}  
)
```



The screenshot shows a terminal window titled 'mongosh mongodb://127.0.0.1'. The command entered is 'db.Scholarship.countDocuments({ ... })'. The output shows the query definition followed by the number '3' indicating the count of documents that match the criteria.

```
mongosh mongodb://127.0.0.1...  
db.Scholarship.countDocuments({  
    Breakdown: { $elemMatch: { $regex: /^Free/ } },  
    "Sponsor.Headquarters": "United Kingdom",  
    "College.City": "Dublin",  
    Discipline: { $size: 3 },  
    iAnnual_Value: { $lt: 10000 }  
})  
3  
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> |
```

Image 7.1.1

Mongo Db Find command

```
db.Scholarship.find(  
{  
    Breakdown: { $elemMatch: { $regex: /^Free/ } },  
    "Sponsor.Headquarters": "United Kingdom",  
    "College.City": "Dublin",  
    Discipline: { $size: 3 },  
    iAnnual_Value: { $lt: 10000 }  
},  
{  
    Title: 1,  
    Breakdown: 1,  
    Discipline: 1,  
    "College.Name": 1,  
    "College.City": 1,  
    "Sponsor.Headquarters": 1,  
    "Sponsor.Name": 1,  
    iAnnual_Value: 1  
})
```

The screenshot shows a MongoDB shell window titled "mongosh mongodb://127.0.0.1:27017". The command entered is a MongoDB query to find documents in the "db.Scholarship" collection. The query uses \$elemMatch to filter documents where the "Breakdown" field contains a value starting with "Free" and has exactly 3 disciplines. It also filters by "Sponsor.Headquarters": "United Kingdom", "College.City": "Dublin", and "iAnnual_Value": { \$lt: 10000 }. The results are displayed as an array of three documents, each containing fields like _id, Title, Discipline, Breakdown, iAnnual_Value, College, and Sponsor.

```

College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.find(
... {
...     Breakdown: { $elemMatch: { $regex: /^Free/ } },
...     "Sponsor.Headquarters": "United Kingdom",
...     "College.City": "Dublin",
...     Discipline: { $size: 3 },
...     iAnnual_Value: { $lt: 10000 }
... },
... {
...     Title: 1,
...     Breakdown: 1,
...         Discipline: 1,
...     "College.Name": 1,
...     "College.City": 1,
...     "Sponsor.Headquarters": 1,
...     "Sponsor.Name": 1,
...     iAnnual_Value: 1
... }
... )
...
[
{
    _id: 37,
    Title: 'EduFund',
    Discipline: [ 'Medicine', 'Engineering', 'Arts' ],
    Breakdown: [ 'Free Books', 'Free Campus Accommodation', 'Laptop Grant' ],
    iAnnual_Value: 9000,
    College: { Name: 'Griffith College Dublin', City: 'Dublin' },
    Sponsor: { Name: 'Eir', Headquarters: 'United Kingdom' }
},
{
    _id: 660,
    Title: 'EduFund',
    Discipline: [ 'Medicine', 'Business', 'Arts' ],
    Breakdown: [ 'Free Fees', 'Additional Tutor Support' ],
    iAnnual_Value: 9000,
    College: { Name: 'Hibernia College', City: 'Dublin' },
    Sponsor: { Name: 'Eir', Headquarters: 'United Kingdom' }
},
{
    _id: 752,
    Title: 'TechStart',
    Discipline: [ 'Medicine', 'Engineering', 'Business' ],
    Breakdown: [
        'Additional Tutor Support',
        'Laptop Grant',
        'Free Fees',
        'Free Campus Accommodation'
    ],
    iAnnual_Value: 9000,
    College: { Name: 'DIT', City: 'Dublin' },
    Sponsor: { Name: 'AIB Group', Headquarters: 'United Kingdom' }
}
]
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>

```

Image 7.2.1

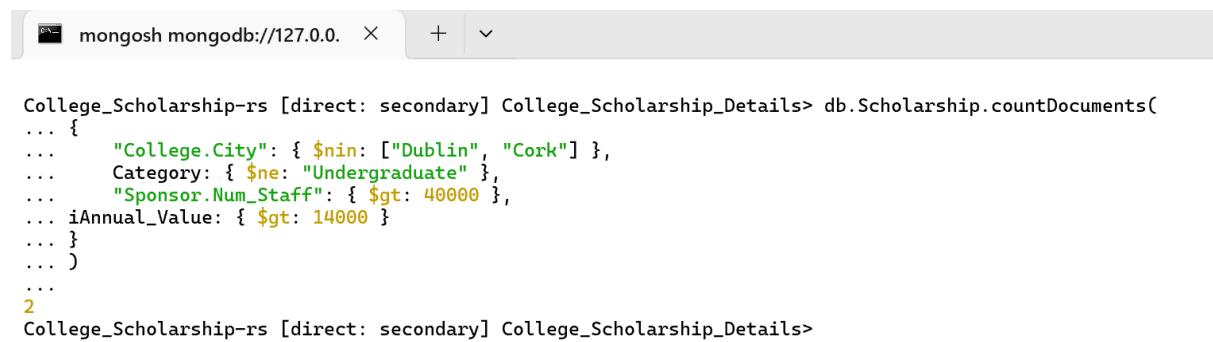
In the above query, we used \$elemMatch with \$regex to match the documents with Breakdown element starting with “Free” and \$size 3 to ensure that there are exactly 3 disciplines. The \$lt for annual values less than €10000, and string match for filtering out Sponsors from UK and Colleges from Dublin.

Query 8

Question: Find the Title, Sponsor Name, College Name whose college is not located in city like **Dublin** or **Cork**, the category should not include **undergraduates**, number of staff should be greater then **35000**, and the annual value should be atleast **14000**.

Mongo Db Count command

```
db.Scholarship.countDocuments(  
{  
    "College.City": { $nin: ["Dublin", "Cork"] },  
    Category: { $ne: "Undergraduate" },  
    "Sponsor.Num_Staff": { $gt: 40000 },  
    iAnnual_Value: { $gt: 14000 }  
}  
)
```



The screenshot shows a terminal window titled 'mongosh mongodb://127.0.0.' with a single tab. The command entered is a MongoDB query to count documents in the 'Scholarship' collection. The query filters for cities not in Dublin or Cork, categories not equal to 'Undergraduate', staff counts greater than 40000, and annual values greater than 14000. The output shows the command and a status message indicating it's running.

```
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.countDocuments(  
... {  
...     "College.City": { $nin: ["Dublin", "Cork"] },  
...     Category: { $ne: "Undergraduate" },  
...     "Sponsor.Num_Staff": { $gt: 40000 },  
...     iAnnual_Value: { $gt: 14000 }  
... }  
... )  
...  
2  
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>
```

Image 8.1.1

Mongo Db Find command

```
db.Scholarship.find(  
{  
    "College.City": { $nin: ["Dublin", "Cork"] },  
    Category: { $ne: "Undergraduate" },  
    "Sponsor.Num_Staff": { $gt: 40000 },  
    iAnnual_Value: { $gt: 14000 }  
},  
{  
    Title: 1,  
    "College.City": 1,  
    "College.Name": 1,  
    "Sponsor.Num_Staff": 1,  
    "Sponsor.Name": 1,  
    Category: 1,  
    iAnnual_Value: 1,  
    _id: 0  
})
```

```

College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.find(
... {
...     "College.City": { $nin: ["Dublin", "Cork"] },
...     "Category": { $ne: "Undergraduate" },
...     "Sponsor.Num_Staff": { $gt: 40000 },
...     "iAnnual_Value": { $gt: 14000 }
... },
... {
...     Title: 1,
...     "College.City": 1,
...     "College.Name": 1,
...     "Sponsor.Num_Staff": 1,
...     "Sponsor.Name": 1,
...     Category: 1,
...     iAnnual_Value: 1,
...     _id: 0
... }
... )
...
[
{
    Title: 'SmartStep',
    iAnnual_Value: 15000,
    Category: 'Postgraduate',
    College: { Name: 'Magee', City: 'Derry (Londonderry)' },
    Sponsor: { Name: 'Ryanair', Num_Staff: 42690 }
},
{
    Title: 'SmartStep',
    iAnnual_Value: 15000,
    Category: 'Postgraduate',
    College: { Name: 'IT Tallaght', City: 'Tallaght (Dublin)' },
    Sponsor: { Name: 'Ryanair', Num_Staff: 42690 }
}
]
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>

```

Image 8.2.1

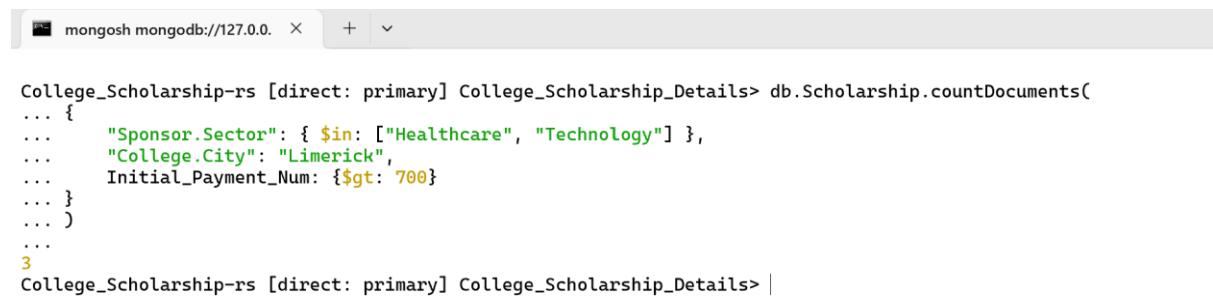
In this query, we used \$nin which is “not in” to exclude colleges from Cities like Dublin and Cork and \$ne which is “not equal” to exclude undergraduate scholarships. Applied \$gt for sponsors staff greater then 40000 and annual values greater then €14000.

Query 9

Question: Display the id, Title, Sponsor Name and college name where Sponsor Sector is in Healthcare or Technology, College is located in Limerick city, and Initial Pay is greater than €700. Sort the results by Initial Pay num Desc

Mongo Db Count command

```
db.Scholarship.countDocuments(  
{  
    "Sponsor.Sector": { $in: ["Healthcare", "Technology"] },  
    "College.City": "Limerick",  
    Initial_Payment_Num: {$gt: 700}  
}  
)
```



The screenshot shows a terminal window titled 'mongosh mongodb://127.0.0.' with a single tab. The command entered is a MongoDB query to count documents in the 'Scholarship' collection. The query filters for 'Sponsor.Sector' in ['Healthcare', 'Technology'], 'College.City' as 'Limerick', and 'Initial_Payment_Num' greater than 700. The output shows the count as 3.

```
College_Scholarship-rs [direct: primary] College_Scholarship_Details> db.Scholarship.countDocuments(  
... {  
...     "Sponsor.Sector": { $in: ["Healthcare", "Technology"] },  
...     "College.City": "Limerick",  
...     Initial_Payment_Num: {$gt: 700}  
... }  
... )  
...  
3  
College_Scholarship-rs [direct: primary] College_Scholarship_Details> |
```

Image 9.1.1

Mongo Db Find command

```
db.Scholarship.find(  
{  
    "Sponsor.Sector": { $in: ["Healthcare", "Technology"] },  
    "College.City": "Limerick",  
    Initial_Payment_Num: {$gt: 700}  
},  
{  
    Title: 1,  
    Initial_Payment_Num: 1,  
    College.City: 1,  
    College.Name: 1,  
    Sponsor.Sector: 1,  
    Sponsor.Name: 1,  
    _id: 0  
}  
) .sort({ Initial_Payment_Num: -1 })
```

```
mongosh mongodb://127.0.0.1:27017/College_Scholarship_rs [direct: primary] College_Scholarship_Details> db.Scholarship.find({  
...   $and:[  
...     {"Sponsor.Sector": { $in: ["Healthcare", "Technology"] }},  
...     {"College.City": "Limerick"},  
...     {"Initial_Payment_Num": {$gt: 700}}  
...   ],  
...   {  
...     Title: 1,  
...     Initial_Payment_Num: 1,  
...     "College.City": 1,  
...     "College.Name": 1,  
...     "Sponsor.Sector": 1,  
...     "Sponsor.Name": 1,  
...     _id: 0  
...   }  
... ).sort({ Initial_Payment_Num: -1 })  
[  
  {  
    Title: 'InspireU',  
    Initial_Payment_Num: 968,  
    College: { Name: 'Limerick IT', City: 'Limerick' },  
    Sponsor: { Name: 'Intel', Sector: 'Technology' }  
  },  
  {  
    Title: 'FutureAid',  
    Initial_Payment_Num: 789,  
    College: { Name: 'UL', City: 'Limerick' },  
    Sponsor: { Name: 'Facebook', Sector: 'Technology' }  
  },  
  {  
    Title: 'StudyNow',  
    Initial_Payment_Num: 722,  
    College: { Name: 'UL', City: 'Limerick' },  
    Sponsor: { Name: 'Pfizer', Sector: 'Healthcare' }  
  }  
]  
College_Scholarship_rs [direct: primary] College_Scholarship_Details> |
```

Image 9.2.1

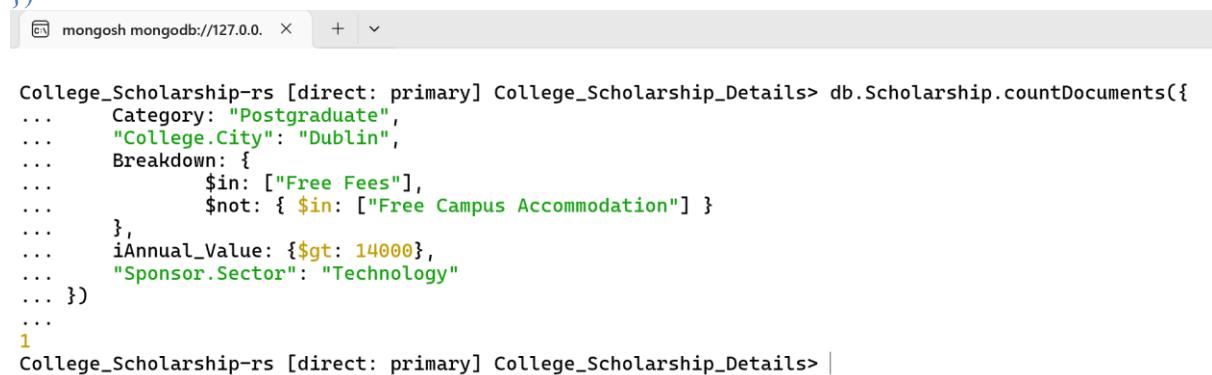
In this query, we have used \$in to match Healthcare and Technology sectors and \$gt for initial payments greater than €700. The sort with -1 to order the results by initial payment in descending order.

Query 10

Question: Find out the **postgraduate** students who are studying in the university located in **Dublin**, that do not include **Free Campus Accommodation** in breakdown section but include **Free Fees** inside the section, have annual fees more then **€14000**, and are sponsored by a company in **Technology** sector.

Mongo Db Count command

```
db.Scholarship.countDocuments({  
    Category: "Postgraduate",  
    "College.City": "Dublin",  
    Breakdown: {  
        $in: ["Free Fees"],  
        $not: { $in: ["Free Campus Accommodation"] }  
    },  
    iAnnual_Value: {$gt: 14000},  
    "Sponsor.Sector": "Technology"  
})
```



The screenshot shows a terminal window titled 'mongosh mongodb://127.0.0.1:27017'. It displays the MongoDB command above, followed by its execution results. The output shows the query being run and then a single digit '1' indicating the count of documents found.

```
College_Scholarship-rs [direct: primary] College_Scholarship_Details> db.Scholarship.countDocuments({  
...     Category: "Postgraduate",  
...     "College.City": "Dublin",  
...     Breakdown: {  
...         $in: ["Free Fees"],  
...         $not: { $in: ["Free Campus Accommodation"] }  
...     },  
...     iAnnual_Value: {$gt: 14000},  
...     "Sponsor.Sector": "Technology"  
... })  
1  
College_Scholarship-rs [direct: primary] College_Scholarship_Details> |
```

Image 10.1.1

Mongo Db Find command

```
db.Scholarship.find({  
    Category: "Postgraduate",  
    "College.City": "Dublin",  
    Breakdown: {  
        $in: ["Free Fees"],  
        $not: { $in: ["Free Campus Accommodation"] }  
    },  
    iAnnual_Value: {$gt: 14000},  
    "Sponsor.Sector": "Technology"  
},{  
    Title: 1,  
    Category: 1,  
    "College.Name": 1,  
    "College.City": 1,  
    Breakdown: 1,  
    iAnnual_Value: 1,  
    "Sponsor.Sector": 1,  
    "Sponsor.Name": 1  
})
```

```

mongosh mongodb://127.0.0.1:27017/College_Scholarship_rs [direct: primary] College_Scholarship_Details> db.Scholarship.find({
...   Category: "Postgraduate",
...   "College.City": "Dublin",
...   Breakdown: {
...     $in: ["Free Fees"],
...     $not: { $in: ["Free Campus Accommodation"] }
...   },
...   iAnnual_Value: {$gt: 14000},
...   "Sponsor.Sector": "Technology"
... }, {
...   Title: 1,
...   Category: 1,
...   "College.Name": 1,
...   "College.City": 1,
...   Breakdown: 1,
...   iAnnual_Value: 1,
...   "Sponsor.Sector": 1,
...   "Sponsor.Name": 1
... })
...
[
  {
    _id: 885,
    Title: 'InspireU',
    Breakdown: [
      'Free Books',
      'Additional Tutor Support',
      'Laptop Grant',
      'Free Fees'
    ],
    iAnnual_Value: 15000,
    Category: 'Postgraduate',
    College: { Name: 'National College Of Ireland', City: 'Dublin' },
    Sponsor: { Name: 'Intel', Sector: 'Technology' }
  }
]
College_Scholarship_rs [direct: primary] College_Scholarship_Details>

```

Image 10.2.1

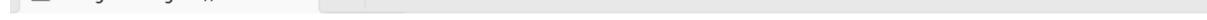
In this query, we used \$in and \$not within Breakdown to include Free Fees but to exclude Free Campus Accommodation for Postgraduate students. We applied \$gt for annual values greater than €14000 and string match for sponsors in Technology Sector.

Query 11

Question: Display id, Title, College name who are **undergraduated** from any of the discipline **Business Or Science** at college with less then **4000** students, who are not sponsored by companies in **Banking or Airlines** and **Manufacturing** sector, which also offers **free campus accommodation** and **additional tutor support**, and have an annual value sponsorship which is less then **€10000**.

Mongo Db Count command

```
db.Scholarship.countDocuments(  
    {  
        $or:  
            [  
                { Discipline: "Business" },  
                { Discipline: "Science" }  
            ],  
            Category: "Undergraduate",  
            "College.Num_Students": { $lt: 4000 },  
            "Sponsor.Sector": { $nin: ["Banking", "Airlines", "Manufacturing"] },  
            Breakdown: { $all: ["Free Campus Accommodation", "Additional Tutor Support"] },  
            iAnnual_Value: { $lt: 10000 }  
    }  
)
```



```
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.countDocuments(  
... {  
... $or:  
... [  
...     { Discipline: "Business" },  
...     { Discipline: "Science" }  
... ],  
... Category: "Undergraduate",  
... "College.Num_Students": { $lt: 4000 },  
... "Sponsor.Sector": { $nin: ["Banking", "Airlines", "Manufacturing"] },  
... Breakdown: { $all: ["Free Campus Accommodation", "Additional Tutor Support"] },  
... iAnnual_Value: { $lt: 10000 }  
... }  
... )  
3  
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>
```

Image 11.1.1

Mongo Db Find command

```
db.Scholarship.find(  
    {  
        $or:  
            [  
                { Discipline: "Business" },  
                { Discipline: "Science" }  
            ],  
            Category: "Undergraduate",  
            "College.Num_Students": { $lt: 4000 },  
            "Sponsor.Sector": { $nin: ["Banking", "Airlines", "Manufacturing"] },  
            Breakdown: { $all: ["Free Campus Accommodation", "Additional Tutor Support"] },  
            iAnnual_Value: { $lt: 10000 }  
    },
```

```

{
  Title: 1,
  "College.Name": 1,
  iAnnual_Value: 1,
  Category: 1,
  Discipline: 1,
  "College.Num_Students": 1,
  "Sponsor.Sector": 1,
  Breakdown: 1
}
)

```

```

mongosh mongodb://127.0.0.1:27017/College_Scholarship_rs [direct: secondary] College_Scholarship_Details> db.Scholarship.find(
... {
...   $or: [
...     { Discipline: "Business" },
...     { Discipline: "Science" }
...   ],
...   Category: "Undergraduate",
...   "College.Num_Students": { $lt: 4000 },
...   "Sponsor.Sector": { $nin: ["Banking", "Airlines", "Manufacturing"] },
...   Breakdown: { $all: ["Free Campus Accommodation", "Additional Tutor Support"] },
...   iAnnual_Value: { $lt: 10000 }
... },
... {
...   Title: 1,
...   "College.Name": 1,
...   iAnnual_Value: 1,
...   Category: 1,
...   Discipline: 1,
...   "College.Num_Students": 1,
...   "Sponsor.Sector": 1,
...   Breakdown: 1
... }
...
[ {
  _id: 27,
  Title: 'SkillRise',
  Discipline: [ 'Medicine', 'Business', 'Arts' ],
  Breakdown: [
    'Free Books',
    'Free Campus Accommodation',
    'Additional Tutor Support'
  ],
  iAnnual_Value: 9000,
  Category: 'Undergraduate',
  College: { Name: 'Athlone IT', Num_Students: 3610 },
  Sponsor: { Sector: 'Manufacturing' }
},
{
  _id: 641,
  Title: 'EduFund',
  Discipline: [ 'Business', 'Arts', 'Engineering' ],
  Breakdown: [
    'Additional Tutor Support',
    'Laptop Grant',
    'Free Campus Accommodation'
  ],
  iAnnual_Value: 9000,
  Category: 'Undergraduate',
  College: { Name: 'UCC', Num_Students: 3460 },
  Sponsor: { Sector: 'Telecommunications' }
},
{
  _id: 652,
  Title: 'SkillRise',
  Discipline: [ 'Science', 'Engineering', 'Medicine', 'Arts' ],
  Breakdown: [
    'Additional Tutor Support',
    'Laptop Grant',
    'Free Campus Accommodation'
  ],
  iAnnual_Value: 9000,
  Category: 'Undergraduate',
  College: { Name: 'Shannon College Of Hotel Mgt', Num_Students: 3730 },
  Sponsor: { Sector: 'Manufacturing' }
}
]

```

Image 11.2.1

```

Discipline: [ 'Business', 'Arts', 'Engineering' ],
Breakdown: [
  'Additional Tutor Support',
  'Laptop Grant',
  'Free Campus Accommodation'
],
iAnnual_Value: 9000,
Category: 'Undergraduate',
College: { Name: 'UCC', Num_Students: 3460 },
Sponsor: { Sector: 'Telecommunications' }
},
{
  _id: 652,
  Title: 'SkillRise',
  Discipline: [ 'Science', 'Engineering', 'Medicine', 'Arts' ],
  Breakdown: [
    'Additional Tutor Support',
    'Laptop Grant',
    'Free Campus Accommodation'
  ],
  iAnnual_Value: 9000,
  Category: 'Undergraduate',
  College: { Name: 'Shannon College Of Hotel Mgt', Num_Students: 3730 },
  Sponsor: { Sector: 'Manufacturing' }
}
]
College_Scholarship_rs [direct: secondary] College_Scholarship_Details>

```

Image 11.2.2

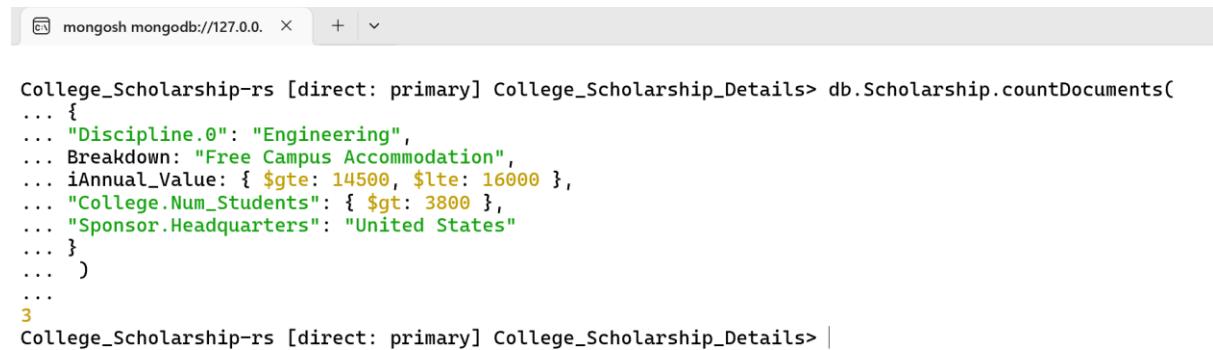
In this query, we used `$or` to match Business or Science discipline, instead of `$or` this can be also achieved by `$in` which is a convenient way to do this, `$nin` excludes Banking, Airlines or Manufacturing sectors. We applied `$all` for Free Campus Accommodation and Additional Tutor Support and `$lt` for annual values less than €10000, and colleges with less than 4000 students.

Query 12

Question: Show Title, College Name for scholarships that have **Engineering** as their first discipline, offer **free campus accommodation**, have an annual value between **€14500** and **€16000**, are at college with more than **3800** students, and are sponsored by companies in **United States**.

Mongo Db Count command

```
db.Scholarship.countDocuments(  
  {  
    "Discipline.0": "Engineering",  
    Breakdown: "Free Campus Accommodation",  
    iAnnual_Value: { $gte: 14500, $lte: 16000 },  
    "College.Num_Students": { $gt: 3800 },  
    "Sponsor.Headquarters": "United States"  
  }  
)
```



The screenshot shows a terminal window titled 'mongosh mongodb://127.0.0.1'. The command entered is a MongoDB query to count documents in the 'Scholarship' collection. The query filters for scholarships where the first discipline is 'Engineering', the breakdown is 'Free Campus Accommodation', the annual value is between €14500 and €16000, the number of students is greater than 3800, and the sponsor's headquarters is 'United States'. The output shows the count of 3 documents.

```
College_Scholarship-rs [direct: primary] College_Scholarship_Details> db.Scholarship.countDocuments(  
... {  
...   "Discipline.0": "Engineering",  
...   Breakdown: "Free Campus Accommodation",  
...   iAnnual_Value: { $gte: 14500, $lte: 16000 },  
...   "College.Num_Students": { $gt: 3800 },  
...   "Sponsor.Headquarters": "United States"  
... }  
... )  
...  
3  
College_Scholarship-rs [direct: primary] College_Scholarship_Details> |
```

Image 12.1.1

Mongo Db Find command

```
db.Scholarship.find(  
  {  
    "Discipline.0": "Engineering",  
    Breakdown: "Free Campus Accommodation",  
    iAnnual_Value: { $gte: 14500, $lte: 16000 },  
    "College.Num_Students": { $gt: 3800 },  
    "Sponsor.Headquarters": "United States"  
  },  
  {  
    Title: 1,  
    Discipline: 1,  
    Breakdown: 1,  
    "College.Num_Students": 1,  
    "College.Name": 1,  
    iAnnual_Value: 1,  
    _id: 0  
  })
```

```

mongosh mongodb://127.0.0.1:27017/College_Scholarship-rs [direct: primary] College_Scholarship_Details> db.Scholarship.find(
... {
... "Discipline.0": "Engineering",
... "Breakdown": "Free Campus Accommodation",
... iAnnual_Value: { $gte: 14500, $lte: 16000 },
... "College.Num_Students": { $gt: 3800 },
... "Sponsor.Headquarters": "United States"
... },
... {
...     Title: 1,
...     Discipline: 1,
...     Breakdown: 1,
...     "College.Num_Students": 1,
...     "College.Name": 1,
...     iAnnual_Value: 1,
...     _id: 0
... }
... )
...
[
{
    Title: 'FutureAid',
    Discipline: [ 'Engineering' ],
    Breakdown: [
        'Free Fees',
        'Free Campus Accommodation',
        'Free Books',
        'Laptop Grant'
    ],
    iAnnual_Value: 15000,
    College: { Name: 'Queens', Num_Students: 4540 }
},
{
    Title: 'StudyNow',
    Discipline: [ 'Engineering', 'Medicine', 'Business', 'Arts' ],
    Breakdown: [
        'Free Books',
        'Free Campus Accommodation',
        'Additional Tutor Support'
    ],
    iAnnual_Value: 15000,
    College: { Name: 'DCU', Num_Students: 3970 }
},
{
    Title: 'InspireU',
    Discipline: [ 'Engineering', 'Arts' ],
    Breakdown: [ 'Free Fees', 'Free Campus Accommodation' ],
    iAnnual_Value: 15000,
    College: { Name: 'Limerick IT', Num_Students: 4880 }
}
]

```

Image 12.2.1

```

],
iAnnual_Value: 15000,
College: { Name: 'DCU', Num_Students: 3970 }
},
{
    Title: 'InspireU',
    Discipline: [ 'Engineering', 'Arts' ],
    Breakdown: [ 'Free Fees', 'Free Campus Accommodation' ],
    iAnnual_Value: 15000,
    College: { Name: 'Limerick IT', Num_Students: 4880 }
}
]
College_Scholarship-rs [direct: primary] College_Scholarship_Details>

```

Image 12.2.2

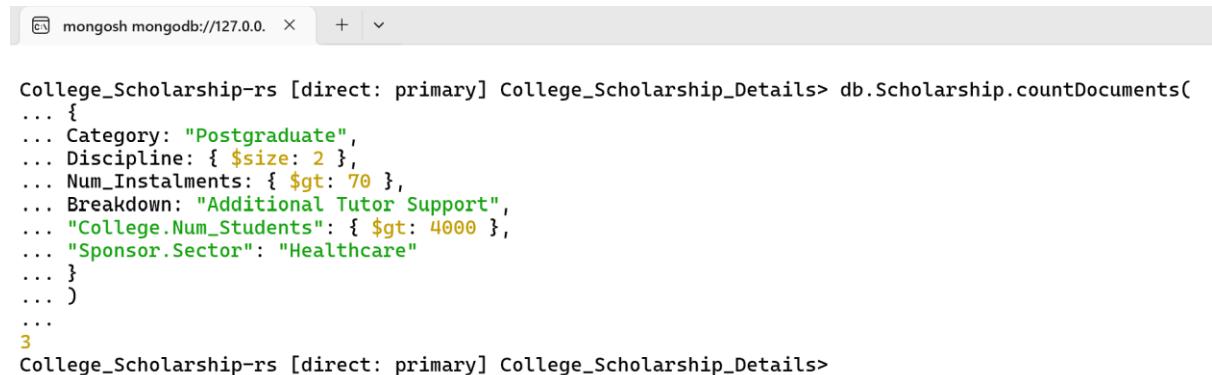
In this query, we used array index (Discipline.0) to match Engineering as the first discipline, string match for Free Campus accommodation. The \$gte and \$lte are used for values between €14500 to €16000, \$gt for colleges with greater than 3800 students and string match for sponsor Headquarters in the US.

Query 13

Question: Display the Title, College Name and Sponsor Name of **postgraduate** scholarship students who have exactly **2 disciplines**, have more than **70** installments, they also offer **additional tutor support** in Breakdown, are studying in college with more than **4000** students, which are sponsored by company in **health care** sector. Sort the results by **number of installments** in Ascending order.

Mongo Db Count command

```
db.Scholarship.countDocuments(  
    {  
        Category: "Postgraduate",  
        Discipline: { $size: 2 },  
        Num_Instalments: { $gt: 70 },  
        Breakdown: "Additional Tutor Support",  
        "College.Num_Students": { $gt: 4000 },  
        "Sponsor.Sector": "Healthcare"  
    }  
)
```



The screenshot shows a terminal window titled 'mongosh mongodb://127.0.0.' with a single tab. The command entered is the MongoDB code provided above. The output shows the command being run and then the result, which is the number '3' indicating there are three documents that match the specified criteria.

```
mongosh mongodb://127.0.0. × + ▾  
  
College_Scholarship-rs [direct: primary] College_Scholarship_Details> db.Scholarship.countDocuments(  
... {  
...     Category: "Postgraduate",  
...     Discipline: { $size: 2 },  
...     Num_Instalments: { $gt: 70 },  
...     Breakdown: "Additional Tutor Support",  
...     "College.Num_Students": { $gt: 4000 },  
...     "Sponsor.Sector": "Healthcare"  
... }  
... }  
...  
3  
College_Scholarship-rs [direct: primary] College_Scholarship_Details>
```

Image 13.1.1

Mongo Db Find command

```
db.Scholarship.find(  
    {  
        Category: "Postgraduate",  
        Discipline: { $size: 2 },  
        Num_Instalments: { $gt: 70 },  
        Breakdown: "Additional Tutor Support",  
        "College.Num_Students": { $gt: 4000 },  
        "Sponsor.Sector": "Healthcare"  
    },  
    {  
        Title: 1,  
        Category: 1,  
        Discipline: 1,  
        Breakdown: 1,  
        "College.Num_Students": 1,  
        "College.Name": 1,  
        "Sponsor.Name": 1,  
        "Sponsor.Sector": 1,  
    }
```

```

Num_Instalments: 1,
_id: 0
}
).sort({ Num_Instalments: 1 })

```

The screenshot shows the MongoDB shell interface with the command `College_Scholarship-rs [direct: primary] College_Scholarship_Details> db.Scholarship.find(` followed by the query code. The results are displayed as an array of documents. The first document has fields: Title: 'StudyNow', Discipline: ['Science', 'Engineering'], Breakdown: ['Free Fees', 'Free Campus Accommodation', 'Additional Tutor Support'], Num_Instalments: 70, Category: 'Postgraduate', College: { Name: 'UL', Num_Students: 4660 }, Sponsor: { Name: 'Pfizer', Sector: 'Healthcare' }. The second document has fields: Title: 'NextGen Grant', Discipline: ['Science', 'Engineering'], Breakdown: ['Free Books', 'Additional Tutor Support'], Num_Instalments: 72, Category: 'Postgraduate', College: { Name: 'UL', Num_Students: 4660 }, Sponsor: { Name: 'Pfizer', Sector: 'Healthcare' }. The third document has fields: Title: 'StudyNow', Discipline: ['Engineering', 'Medicine'], Breakdown: ['Additional Tutor Support', 'Laptop Grant', 'Free Fees', 'Free Campus Accommodation'], Num_Instalments: 80, Category: 'Postgraduate', College: { Name: 'College', Num_Students: 4290 }, Sponsor: { Name: 'Pfizer', Sector: 'Healthcare' }.

```

College_Scholarship-rs [direct: primary] College_Scholarship_Details> db.Scholarship.find(
...
{
  Category: "Postgraduate",
  Discipline: { $size: 2 },
  Num_Instalments: { $gt: 70 },
  Breakdown: "Additional Tutor Support",
  "College.Num_Students": { $gt: 4000 },
  "Sponsor.Sector": "Healthcare"
},
{
  Title: 1,
  Category: 1,
  Discipline: 1,
  Breakdown: 1,
  "College.Num_Students": 1,
  "College.Name": 1,
  "Sponsor.Name": 1,
  "Sponsor.Sector": 1,
  Num_Instalments: 1,
  _id: 0
}
).sort({ Num_Instalments: 1 })

[
{
  Title: 'StudyNow',
  Discipline: [ 'Science', 'Engineering' ],
  Breakdown: [
    'Free Fees',
    'Free Campus Accommodation',
    'Additional Tutor Support'
  ],
  Num_Instalments: 70,
  Category: 'Postgraduate',
  College: { Name: 'UL', Num_Students: 4660 },
  Sponsor: { Name: 'Pfizer', Sector: 'Healthcare' }
},
{
  Title: 'NextGen Grant',
  Discipline: [ 'Science', 'Engineering' ],
  Breakdown: [
    'Free Books',
    'Additional Tutor Support',
    ...
]

```

Image 13.2.1

The screenshot shows the MongoDB shell interface with the command `College_Scholarship-rs [direct: primary] College_Scholarship_Details>` followed by the query code. The results are displayed as an array of documents. The first document has fields: Title: 'StudyNow', Discipline: ['Engineering', 'Medicine'], Breakdown: ['Additional Tutor Support', 'Laptop Grant', 'Free Fees', 'Free Campus Accommodation'], Num_Instalments: 80, Category: 'Postgraduate', College: { Name: 'College', Num_Students: 4290 }, Sponsor: { Name: 'Pfizer', Sector: 'Healthcare' }. The second document has fields: Title: 'NextGen Grant', Discipline: ['Science', 'Engineering'], Breakdown: ['Free Books', 'Additional Tutor Support', 'Laptop Grant', 'Free Fees', 'Free Campus Accommodation'], Num_Instalments: 75, Category: 'Postgraduate', College: { Name: 'National College Of Ireland', Num_Students: 4590 }, Sponsor: { Name: 'DCC plc', Sector: 'Healthcare' }. The third document has fields: Title: 'StudyNow', Discipline: ['Engineering', 'Medicine'], Breakdown: ['Additional Tutor Support', 'Laptop Grant', 'Free Fees', 'Free Campus Accommodation'], Num_Instalments: 80, Category: 'Postgraduate', College: { Name: 'UL', Num_Students: 4660 }, Sponsor: { Name: 'Pfizer', Sector: 'Healthcare' }.

```

College_Scholarship-rs [direct: primary] College_Scholarship_Details>
},
{
  Title: 'NextGen Grant',
  Discipline: [ 'Science', 'Engineering' ],
  Breakdown: [
    'Free Books',
    'Additional Tutor Support',
    'Laptop Grant',
    'Free Campus Accommodation'
  ],
  Num_Instalments: 75,
  Category: 'Postgraduate',
  College: { Name: 'National College Of Ireland', Num_Students: 4590 },
  Sponsor: { Name: 'DCC plc', Sector: 'Healthcare' }
},
{
  Title: 'StudyNow',
  Discipline: [ 'Engineering', 'Medicine' ],
  Breakdown: [
    'Additional Tutor Support',
    'Laptop Grant',
    'Free Fees',
    'Free Campus Accommodation'
  ],
  Num_Instalments: 80,
  Category: 'Postgraduate',
  College: { Name: 'College', Num_Students: 4290 },
  Sponsor: { Name: 'Pfizer', Sector: 'Healthcare' }
}
]
```

Image 13.2.2

In this query, we used \$size to match exactly 2 discipline, string match for Breakdown as Additional Tutor Support. The \$gt for installments greater than 70 and colleges with students more than 4000. The sort makes the order sorted as per Number of Installments in Ascending Order.

Query 14

Question: Find out all the scholarships which are not located in **Dublin**, that also offer **free fees** and either **laptop grant** or **free campus accommodation**, are sponsored by **non-Irish** companies, and have Initial Payment over **€930**.

Mongo Db Count command

```
db.Scholarship.countDocuments({  
    "College.City": { $ne: "Dublin" },  
    $and: [ { Breakdown: "Free Fees" },  
    { $or: [ { Breakdown: "Laptop Grant" },  
    { Breakdown: "Free Campus Accommodation" } ] } ],  
    "Sponsor.Headquarters": { $ne: "Ireland" },  
    Initial_Payment_Num: { $gt: 930 }  
})  
mongosh mongodb://127.0.0.1:27017  
+ v  
  
College_Scholarship-rs [direct: primary] College_Scholarship_Details> db.Scholarship.countDocuments({  
... "College.City": { $ne: "Dublin" },  
... $and: [ { Breakdown: "Free Fees" },  
... { $or: [ { Breakdown: "Laptop Grant" },  
... { Breakdown: "Free Campus Accommodation" } ] } ],  
... "Sponsor.Headquarters": { $ne: "Ireland" },  
... Initial_Payment_Num: { $gt: 930 }  
... } )  
...  
2  
College_Scholarship-rs [direct: primary] College_Scholarship_Details>
```

Image 14.1.1

Mongo Db Find command

```
db.Scholarship.find({  
    "College.City": { $ne: "Dublin" },  
    $and: [ { Breakdown: "Free Fees" },  
    { $or: [ { Breakdown: "Laptop Grant" },  
    { Breakdown: "Free Campus Accommodation" } ] } ],  
    "Sponsor.Headquarters": { $ne: "Ireland" },  
    Initial_Payment_Num: { $gt: 930 }  
},  
{  
    Title: 1,  
    "College.Name": 1,  
    "College.City": 1,  
    Breakdown: 1,  
    "Sponsor.Name": 1,  
    "Sponsor.Headquarters": 1,  
    Initial_Payment_Num: 1  
})
```



```

mongosh mongodb://127.0.0.1:27017/College_Scholarship_rs [direct: primary] College_Scholarship_Details> db.Scholarship.find({
...   "College.City": { $ne: "Dublin" },
...   $and: [ { Breakdown: "Free Fees" },
...   { $or: [ { Breakdown: "Laptop Grant" },
...   { Breakdown: "Free Campus Accommodation" } ] } ],
...   "Sponsor.Headquarters": { $ne: "Ireland" },
...   Initial_Payment_Num: { $gt: 930 }
... },
... {
...   Title: 1,
...   "College.Name": 1,
...   "College.City": 1,
...   Breakdown: 1,
...   "Sponsor.Name": 1,
...   "Sponsor.Headquarters": 1,
...   Initial_Payment_Num: 1
... }
... )
...
[
  {
    _id: 645,
    Title: 'STEM Boost',
    Breakdown: [
      'Free Fees',
      'Free Campus Accommodation',
      'Free Books',
      'Additional Tutor Support'
    ],
    Initial_Payment_Num: 984,
    College: { Name: 'IT Tralee', City: 'Tralee' },
    Sponsor: { Name: 'Permanent TSB', Headquarters: 'Germany' }
  },
  {
    _id: 904,
    Title: 'InspireU',
    Breakdown: [ 'Free Fees', 'Free Campus Accommodation' ],
    Initial_Payment_Num: 968,
    College: { Name: 'Limerick IT', City: 'Limerick' },
    Sponsor: { Name: 'Intel', Headquarters: 'United States' }
  }
]
College_Scholarship_rs [direct: primary] College_Scholarship_Details>

```

Image 14.2.1

In this query, we used \$ne to exclude the colleges from Dublin, \$and with \$or to match Free Fees and either Laptop Grant or Free Campus Accommodation but not both. The \$ne for non Irish Sponsors and \$gt to retrieve initial payments greater than €930.

Query 15

Question: Display the Title, id, College Name and Sector Name where undergraduate scholarship offers exactly two benefits that are not **free fees** or **free books**, have a duration of exactly **4** years, are sponsored by an **airlines** company which are studying at a college in City **Sligo**.

Mongo Db Count command

```
db.Scholarship.countDocuments( {  
    Category: "Undergraduate",  
    Breakdown: {  
        $size: 2,  
        $nin: ["Free Fees", "Free Books"] },  
    Duration: 4,  
    "Sponsor.Sector": "Airlines",  
    "College.City": "Sligo"  
})
```



```
mongosh mongodb://127.0.0.1:27017  
+ v  
College_Scholarship-rs [direct: primary] College_Scholarship_Details> db.Scholarship.countDocuments( {  
...     Category: "Undergraduate",  
...     Breakdown: {  
...         $size: 2,  
...         $nin: ["Free Fees", "Free Books"] },  
...         Duration: 4,  
...         "Sponsor.Sector": "Airlines",  
...         "College.City": "Sligo"  
...     } )  
...  
1  
College_Scholarship-rs [direct: primary] College_Scholarship_Details>
```

Image 15.1.1

Mongo Db Find command

```
db.Scholarship.find( {  
    Category: "Undergraduate",  
    Breakdown: {  
        $size: 2,  
        $nin: ["Free Fees", "Free Books"] },  
    Duration: 4,  
    "Sponsor.Sector": "Airlines",  
    "College.City": "Sligo"  
}, {  
    Title: 1,  
    Category: 1,  
    Breakdown: 1,  
    Duration: 1,  
    "Sponsor.Name": 1,  
    "Sponsor.Sector": 1,  
    "College.Name": 1,  
    "College.City": 1  
})
```

```
mongosh mongodb://127.0.0.1:27017/College_Scholarship_rs [direct: primary] College_Scholarship_Details> db.Scholarship.find( {  
...   Category: "Undergraduate",  
...   Breakdown: {  
...     $size: 2,  
...     $nin: ["Free Fees", "Free Books"] },  
...   Duration: 4,  
...   "Sponsor.Sector": "Airlines",  
...   "College.City": "Sligo"  
... }, {  
...   Title: 1,  
...   Category: 1,  
...   Breakdown: 1,  
...   Duration: 1,  
...   "Sponsor.Name": 1,  
...   "Sponsor.Sector": 1,  
...   "College.Name": 1,  
...   "College.City": 1  
... })  
...  
[  
{  
  _id: 421,  
  Title: 'SmartStep',  
  Breakdown: [ 'Free Campus Accommodation', 'Additional Tutor Support' ],  
  Duration: 4,  
  Category: 'Undergraduate',  
  College: { Name: 'St Angelas', City: 'Sligo' },  
  Sponsor: { Name: 'Ryanair', Sector: 'Airlines' }  
}  
]  
College_Scholarship_rs [direct: primary] College_Scholarship_Details>
```

Image 15.2.1

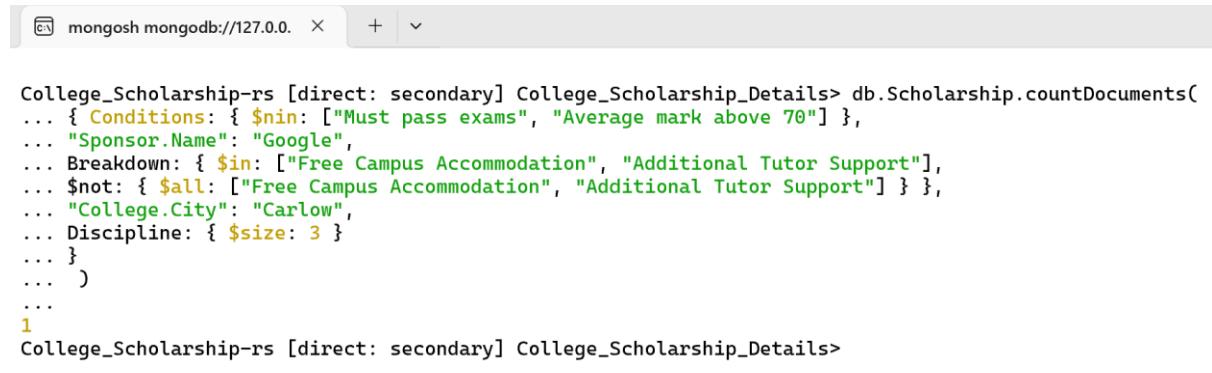
In this query, we used `$size` to match the array size for exactly 2 benefits in `Breakdown`, `$nin` to exclude Free Fees or Free Books. The string match is applied for Sponsors from Airlines sector, duration of scholarship and for selecting undergraduate students.

Query 16

Question: Display the id of those scholarships which does not require **passing exams** nor an **average mark above 70**, are sponsored by company called **Google**, offer either **free campus accommodation** or **additional tutor support** (but not both), are at colleges in **Carlow**, and have exactly three disciplines.

Mongo Db Count command

```
db.Scholarship.countDocuments(  
  { Conditions: { $nin: ["Must pass exams", "Average mark above 70"] },  
    "Sponsor.Name    Breakdown: { $in: ["Free Campus Accommodation", "Additional Tutor Support"],  
      $not: { $all: ["Free Campus Accommodation", "Additional Tutor Support"] } },  
    "College.City    Discipline: { $size: 3 }  
  }  
)
```



```
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.countDocuments(  
... { Conditions: { $nin: ["Must pass exams", "Average mark above 70"] },  
...   "Sponsor.Name...   Breakdown: { $in: ["Free Campus Accommodation", "Additional Tutor Support"],  
...     $not: { $all: ["Free Campus Accommodation", "Additional Tutor Support"] } },  
...   "College.City...   Discipline: { $size: 3 }  
... }  
... )  
...  
1  
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>
```

Image 16.1.1

Mongo Db Find command

```
db.Scholarship.find( {  
  Conditions: { $nin: ["Must pass exams", "Average mark above 70"] },  
  "Sponsor.Name  Breakdown: { $in: ["Free Campus Accommodation", "Additional Tutor Support"],  
    $not: { $all: ["Free Campus Accommodation", "Additional Tutor Support"] } },  
  "College.City  Discipline: { $size: 3 }  
}, {  
  Title: 1,  
  Conditions: 1,  
  Breakdown: 1,  
  "College.City  Discipline: 1  
)
```

```
mongosh mongodb://127.0.0.1:27017/College_Scholarship_rs [direct: secondary] College_Scholarship_Details> db.Scholarship.find( {  
... Conditions: { $nin: ["Must pass exams", "Average mark above 70"] },  
... "Sponsor.Name": "Google",  
... Breakdown: { $in: ["Free Campus Accommodation", "Additional Tutor Support"],  
... $not: { $all: ["Free Campus Accommodation", "Additional Tutor Support"] } },  
... "College.City": "Carlow",  
... Discipline: { $size: 3 }  
... },  
... Title: 1,  
... Conditions: 1,  
... Breakdown: 1,  
... "College.City": 1,  
... Discipline: 1  
... } )  
...  
[  
{  
    _id: 706,  
    Title: 'HopeSchol',  
    Discipline: [ 'Science', 'Medicine', 'Business' ],  
    Breakdown: [ 'Free Fees', 'Additional Tutor Support', 'Laptop Grant' ],  
    Conditions: 'Reviewed at the end of each year',  
    College: { City: 'Carlow' }  
}  
]  
College_Scholarship_rs [direct: secondary] College_Scholarship_Details>
```

Image 16.2.1

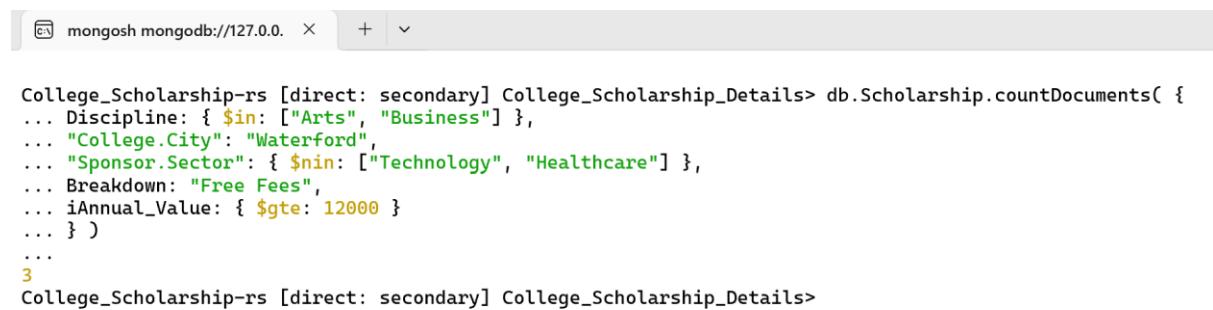
In this query, we have used \$nin to exclude conditions like Must pass exams or Average mark above 70, \$not and \$all to match Free Campus Accommodation or Additional Tutor Support but not both. The \$size is applied for 3 disciplines and string match for Colleges in Carlow City.

Query 17

Question: Find the id, Title for those scholarships where the disciplines are either **Arts** or **Business**, college is located in **Waterford**, the scholarship should not be sponsored by the companies in **Technology** or **Healthcare** sector, offer **free fees** and have sponsorship of atleast **12000**. Sort the Results by **annual value** in desc.

Mongo Db Count command

```
db.Scholarship.countDocuments( {
    Discipline: { $in: ["Arts", "Business"] },
    "College.City": "Waterford",
    "Sponsor.Sector": { $nin: ["Technology", "Healthcare"] },
    Breakdown: "Free Fees",
    iAnnual_Value: { $gte: 12000 }
})
```



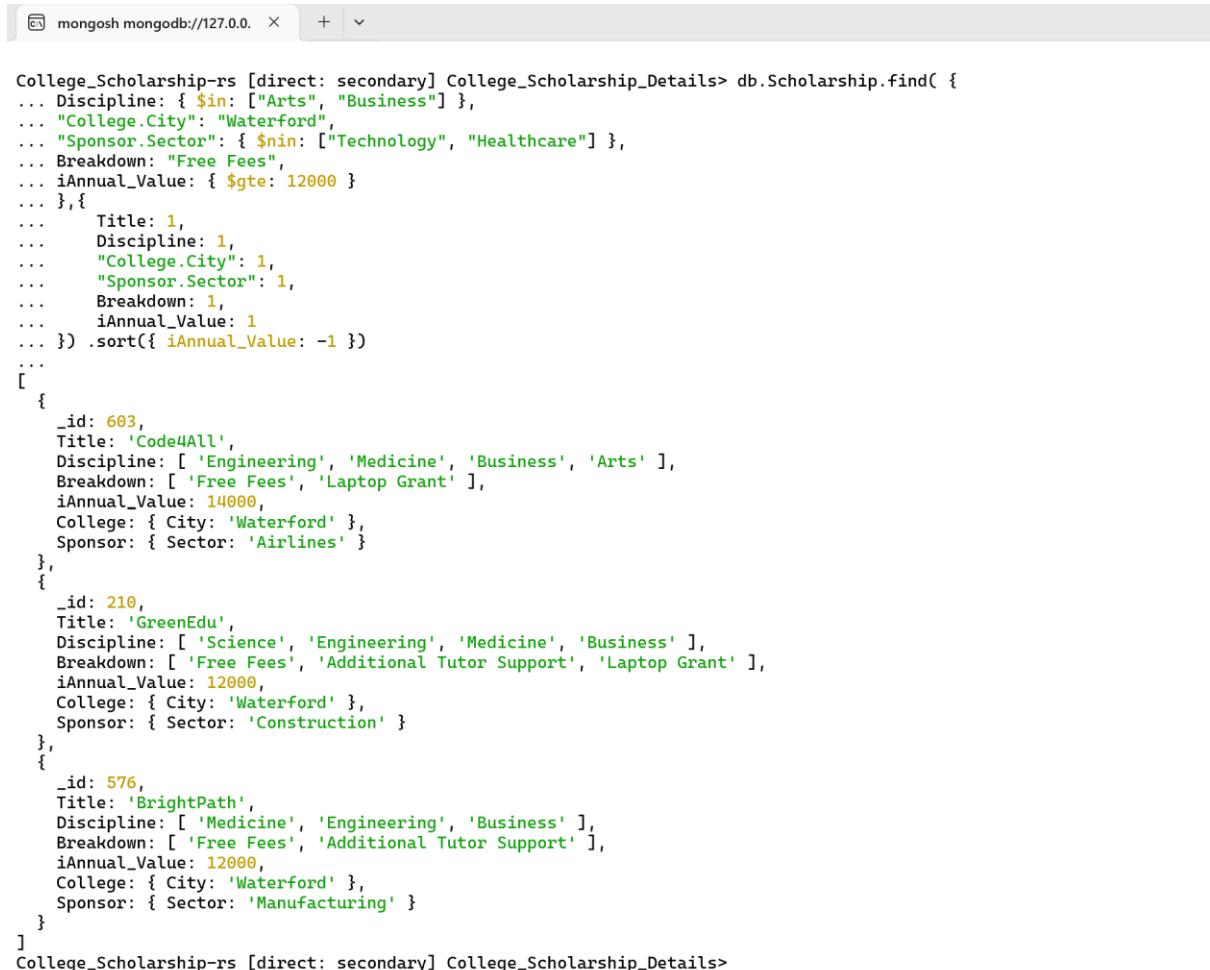
The screenshot shows a terminal window titled 'mongosh mongodb://127.0.0.' with a single tab open. The command entered is a MongoDB query using the `countDocuments` method on the `Scholarship` collection. The query filters for documents where the `Discipline` is either `"Arts"` or `"Business"`, the `College.City` is `"Waterford"`, the `Sponsor.Sector` is not in the array `["Technology", "Healthcare"]`, the `Breakdown` is `"Free Fees"`, and the `iAnnual_Value` is greater than or equal to `12000`. The output shows the result of the query, which is the number 3.

```
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.countDocuments( {
... Discipline: { $in: ["Arts", "Business"] },
... "College.City": "Waterford",
... "Sponsor.Sector": { $nin: ["Technology", "Healthcare"] },
... Breakdown: "Free Fees",
... iAnnual_Value: { $gte: 12000 }
... } )
...
3
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>
```

Image 17.1.1

Mongo Db Find command

```
db.Scholarship.find( {
    Discipline: { $in: ["Arts", "Business"] },
    "College.City": "Waterford",
    "Sponsor.Sector": { $nin: ["Technology", "Healthcare"] },
    Breakdown: "Free Fees",
    iAnnual_Value: { $gte: 12000 }
}, {
    Title: 1,
    Discipline: 1,
    "College.City": 1,
    "Sponsor.Sector": 1,
    Breakdown: 1,
    iAnnual_Value: 1
}).sort({ iAnnual_Value: -1 })
```



```

mongosh mongodb://127.0.0.1:27017/College_Scholarship_rs [direct: secondary] College_Scholarship_Details> db.Scholarship.find( {
... Discipline: { $in: ["Arts", "Business"] },
... "College.City": "Waterford",
... "Sponsor.Sector": { $nin: ["Technology", "Healthcare"] },
... Breakdown: "Free Fees",
... iAnnual_Value: { $gte: 12000 }
... },{
...     Title: 1,
...     Discipline: 1,
...     "College.City": 1,
...     "Sponsor.Sector": 1,
...     Breakdown: 1,
...     iAnnual_Value: 1
... }) .sort({ iAnnual_Value: -1 })
[
{
    _id: 603,
    Title: 'Code4All',
    Discipline: [ 'Engineering', 'Medicine', 'Business', 'Arts' ],
    Breakdown: [ 'Free Fees', 'Laptop Grant' ],
    iAnnual_Value: 14000,
    College: { City: 'Waterford' },
    Sponsor: { Sector: 'Airlines' }
},
{
    _id: 210,
    Title: 'GreenEdu',
    Discipline: [ 'Science', 'Engineering', 'Medicine', 'Business' ],
    Breakdown: [ 'Free Fees', 'Additional Tutor Support', 'Laptop Grant' ],
    iAnnual_Value: 12000,
    College: { City: 'Waterford' },
    Sponsor: { Sector: 'Construction' }
},
{
    _id: 576,
    Title: 'BrightPath',
    Discipline: [ 'Medicine', 'Engineering', 'Business' ],
    Breakdown: [ 'Free Fees', 'Additional Tutor Support' ],
    iAnnual_Value: 12000,
    College: { City: 'Waterford' },
    Sponsor: { Sector: 'Manufacturing' }
}
]
College_Scholarship_rs [direct: secondary] College_Scholarship_Details>

```

Image 17.2.1

In this query, we used `$in` to match Arts or Business, `$nin` to exclude Technology or Healthcare sectors. We have applied string match for Free Fees and `$gte` for annual values greater than equal to 12000. Sorting the values by Annual Values in descending order.

Section 3: Mongo Shell – Aggregation Framework

Aggregation Query 1

Question: What is the count of **Postgraduate** students and what is the total sponsorship annual amount, grouped by different **sponsor names** and sorted by **total annual value** in desc.

Aggregate Query

```
db.Scholarship.aggregate([
  { $match: { Category: "Postgraduate" } },
  { $group: { _id: "$Sponsor.Name",
    Sum_Annual_Value: { $sum: "$iAnnual_Value" },
    Students_Count: { $sum: 1 } },
  { $sort: { Total_Annual_Value: -1 } }
])
```



The screenshot shows the mongo shell interface with the command line 'mongosh mongodb://127.0.0.' and a single tab open. The command entered is the aggregation query provided above. The output shows the results of the aggregation, which is a list of sponsor names with their corresponding annual values and student counts. The results are sorted by total annual value in descending order. The output is as follows:

```
College_Scholarship-rs [direct: primary] College_Scholarship_Details> db.Scholarship.aggregate([
... { $match: { Category: "Postgraduate" } },
... { $group: { _id: "$Sponsor.Name",
...   Sum_Annual_Value: { $sum: "$iAnnual_Value" },
...   Students_Count: { $sum: 1 } },
... { $sort: { Total_Annual_Value: -1 } }
...
[ { _id: 'Irish Life', Sum_Annual_Value: 345000, Students_Count: 31 },
{ _id: 'Facebook', Sum_Annual_Value: 390000, Students_Count: 33 },
{ _id: 'Eir', Sum_Annual_Value: 362000, Students_Count: 29 },
{ _id: 'Aer Lingus', Sum_Annual_Value: 542000, Students_Count: 45 },
{ _id: 'Google', Sum_Annual_Value: 388000, Students_Count: 32 },
{ _id: 'AIB Group', Sum_Annual_Value: 288000, Students_Count: 25 },
{ _id: 'Intel', Sum_Annual_Value: 544000, Students_Count: 46 },
{ _id: 'Pfizer', Sum_Annual_Value: 374000, Students_Count: 30 },
{
  _id: 'Smurfit Kappa',
  Sum_Annual_Value: 386000,
  Students_Count: 32
},
{ '_id: 'Ryanair', Sum_Annual_Value: 381000, Students_Count: 32 },
{ '_id: 'CRH plc', Sum_Annual_Value: 323000, Students_Count: 28 },
{ '_id: 'Glen Dimplex', Sum_Annual_Value: 413000, Students_Count: 34 },
{
  _id: 'Permanent TSB',
  Sum_Annual_Value: 355000,
  Students_Count: 30
},
{ '_id: 'Greencore', Sum_Annual_Value: 386000, Students_Count: 30 },
{ '_id: 'DCC plc', Sum_Annual_Value: 463000, Students_Count: 38 }
]
College_Scholarship-rs [direct: primary] College_Scholarship_Details> |
```

Screenshot 1.1

In the above aggregation query, we have used \$match to filter out Postgraduate students, \$sum to aggregate it by sponsor name and \$sort with -1 to order by total annual value descending, yielding student count and total sponsorship amount.

Aggregation Query 2

Question: Show the Minimum, Maximum and Average Installments from the College that are located in **Cork** or **Athlone**, who have discipline in **Engineering**. Sort the results by Average **Installments** in Ascending.

Aggregate Query

```
db.Scholarship.aggregate([
  { $match: { "College.City": { $in: ["Cork", "Athlone"] } } },
  { $unwind: "$Discipline" },
  { $match: { Discipline: "Engineering" } },
  { $group: { _id: "$College.Name",
    Min_Installments: { $min: "$Num_Instalments" },
    Max_Installments: { $max: "$Num_Instalments" },
    Avg_Installments: { $avg: "$Num_Instalments" } } },
  { $sort: { Avg_Installments: 1 } }
])
```



The screenshot shows a MongoDB shell window titled 'mongosh mongodb://127.0.0...'. The command entered is the aggregation query provided above. The output shows the results for three colleges: Athlone IT, Cork IT, and UCC, each with their respective minimum, maximum, and average installments.

```
College_Scholarship-rs [direct: primary] College_Scholarship_Details> db.Scholarship.aggregate([
... { $match: { "College.City": { $in: ["Cork", "Athlone"] } } },
... { $unwind: "$Discipline" },
... { $match: { Discipline: "Engineering" } },
... { $group: { _id: "$College.Name",
...   Min_Installments: { $min: "$Num_Instalments" },
...   Max_Installments: { $max: "$Num_Instalments" },
...   Avg_Installments: { $avg: "$Num_Instalments" } } },
... { $sort: { Avg_Installments: 1 } }
... ])
[ {
  _id: 'Athlone IT',
  Min_Installments: 60,
  Max_Installments: 84,
  Avg_Installments: 72.38461538461539
},
{
  _id: 'Cork IT',
  Min_Installments: 61,
  Max_Installments: 86,
  Avg_Installments: 72.875
},
{
  _id: 'UCC',
  Min_Installments: 63,
  Max_Installments: 90,
  Avg_Installments: 78.4375
}
]
College_Scholarship-rs [direct: primary] College_Scholarship_Details>
```

Screenshot 2.1

In this query we have used \$match to select colleges from Cork and Athlone, \$unwind to expand the discipline array, and \$match again to filter out Engineering Discipline which was expanded by \$unwind. We have used \$group with \$min, \$max, \$avg to compute installments stats as per college, sorted by average installments in ascending order.

Aggregation Query 3

Question: Calculate the count of scholarship students grouped by **Sponsor name** and category as **High** and **Low**, but exclude the sponsors from sector **Technology, Banking, Manufacturing, Airlines**.

A scholarship is high if **Annual Value** is greater than 12000 or else it should be low. Sort the results by **Sponsor names** in ascending order.

Aggregate Query

```
db.Scholarship.aggregate([
  { $match: { "Sponsor.Sector": { $nin: ["Technology", "Banking", "Manufacturing",
  "Airlines"] } } },
  { $project: {
    "Sponsor.Name": 1,
    Category: { $cond: {
      if: { $gt: ["$iAnnual_Value", 12000] },
      then: "High", else: "Low" } } } },
  { $group: {
    _id: { valueCategory: "$Category",
    Sector_Name: "$Sponsor.Name" },
    Count_Students: { $sum: 1 } } },
  { $sort: { "_id.Sector_Name": 1 } }
])
```



The screenshot shows a terminal window titled 'mongosh mongodb://127.0.0.' with the command 'College_Scholarship_rs [direct: primary] College_Scholarship_Details> db.Scholarship.aggregate([... { \$match: { "Sponsor.Sector": { \$nin: ["Technology", "Banking", "Manufacturing", "Airlines"] } } }, { \$project: { "Sponsor.Name": 1, Category: { \$cond: { if: { \$gt: ["\$iAnnual_Value", 12000] }, then: "High", else: "Low" } } } }, { \$group: { _id: { valueCategory: "\$Category", Sector_Name: "\$Sponsor.Name" }, Count_Students: { \$sum: 1 } } }, { \$sort: { "_id.Sector_Name": 1 } }])'. The output displays the results of the aggregation, showing 10 documents with fields '_id.valueCategory', '_id.Sector_Name', and 'Count_Students'.

_id.valueCategory	_id.Sector_Name	Count_Students
'Low'	'CRH plc'	44
'High'	'CRH plc'	29
'Low'	'DCC plc'	34
'High'	'DCC plc'	31
'High'	'Eir'	30
'Low'	'Eir'	34
'High'	'Pfizer'	21
'Low'	'Pfizer'	33

Screenshot 3.1

In this query, \$match with \$nin to exclude Technology, Banking and Manufacturing, and Airlines sponsor, and \$project with \$cond to categorise scholarships as High (> 12000) or Low. We applied \$group with \$sum to count students per sponsor and category, sorted by sponsor name in ascending.

Aggregation Query 4

Question: Calculate the total sponsorship amount by multiplying the **num_installments** and **initial payment num**, to compute total scholarship amount. Then group the results by **total sponsorship amount** by sector and number of sponsored students. Finally sort the results by **total sponsorship amount** in desc.

Aggregate Query

```
db.Scholarship.aggregate([
  { $addFields: {
    Total_Scholarship: { $multiply: ["$Num_Instalments", "$Initial_Payment_Num"] } },
  { $group: { _id: "$Sponsor.Sector",
    Total_Scholarship_Amount: { $sum: "$Total_Scholarship" },
    Scholarship_Count: { $sum: 1 } },
  { $sort: { Total_Sponsor_Amount: -1 } }
])
```



The screenshot shows the MongoDB shell interface with the command `db.Scholarship.aggregate` and its output. The output displays the results grouped by sector, showing the total scholarship amount and the count of sponsored students for each sector.

```
College_Scholarship-rs [direct: primary] College_Scholarship_Details> db.Scholarship.aggregate([
...   { $addFields: {
...     Total_Scholarship: { $multiply: ["$Num_Instalments", "$Initial_Payment_Num"] } },
...   { $group: { _id: "$Sponsor.Sector",
...     Total_Scholarship_Amount: { $sum: "$Total_Scholarship" },
...     Scholarship_Count: { $sum: 1 } },
... { $sort: { Total_Sponsor_Amount: -1 } }
... ])
...
[{
  _id: 'Technology',
  Total_Scholarship_Amount: 8730243,
  Scholarship_Count: 207
},
{
  _id: 'Airlines',
  Total_Scholarship_Amount: 6204519,
  Scholarship_Count: 147
},
{
  _id: 'Telecommunications',
  Total_Scholarship_Amount: 2782009,
  Scholarship_Count: 64
},
{
  _id: 'Construction',
  Total_Scholarship_Amount: 3000427,
  Scholarship_Count: 73
},
{
  _id: 'Banking',
  Total_Scholarship_Amount: 7888947,
  Scholarship_Count: 190
},
{
  _id: 'Manufacturing',
  Total_Scholarship_Amount: 8520544,
  Scholarship_Count: 200
},
{
  _id: 'Healthcare',
  Total_Scholarship_Amount: 5174905,
  Scholarship_Count: 119
}]
```

College_Scholarship-rs [direct: primary] College_Scholarship_Details>

Screenshot 4.1

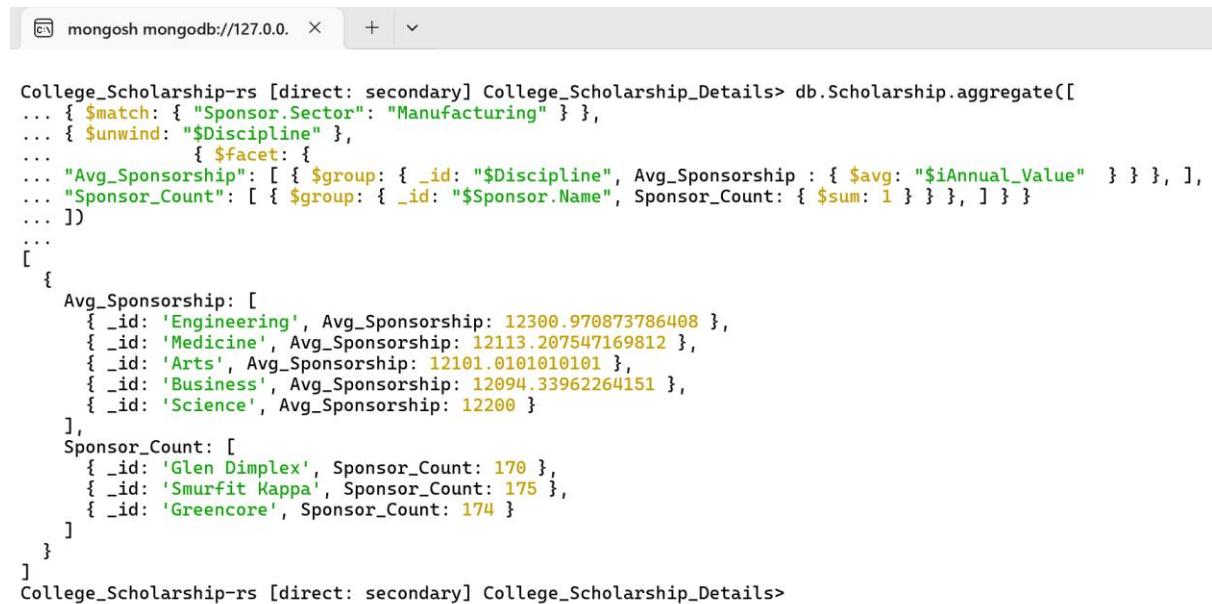
In this query, `$addFields` with `$multiply` to compute total scholarship amount of installments and initial payments, and `$group` with `$sum` to aggregate by sector and `$sort` with `-1` to order the total scholarship amount in descending.

Aggregation Query 5

Question: What is the average annual sponsorship amount for each **discipline** in scholarships sponsored by the **manufacturing** sector, and how many scholarships has each sponsor provided?

Aggregate Query

```
db.Scholarship.aggregate([
  { $match: { "Sponsor.Sector": "Manufacturing" } },
  { $unwind: "$Discipline" },
  { $facet: {
    "Avg_Sponsorship": [ { $group: { _id: "$Discipline", Avg_Sponsorship : { $avg: "$iAnnual_Value" } } }, ],
    "Sponsor_Count": [ { $group: { _id: "$Sponsor.Name", Sponsor_Count: { $sum: 1 } } }, ]
  } }
])
```



The screenshot shows a MongoDB shell window titled 'mongosh mongodb://127.0.0.1:27017'. The command entered is the aggregation query from above. The output displays the results of the aggregation, which include two main facets: 'Avg_Sponsorship' and 'Sponsor_Count'. The 'Avg_Sponsorship' facet shows the average sponsorship amount for each discipline, with values ranging from approximately 12101 to 12300. The 'Sponsor_Count' facet shows the count of scholarships provided by each sponsor, with three sponsors having counts of 170, 175, and 174 respectively.

```
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.aggregate([
... { $match: { "Sponsor.Sector": "Manufacturing" } },
... { $unwind: "$Discipline" },
... { $facet: {
...   "Avg_Sponsorship": [ { $group: { _id: "$Discipline", Avg_Sponsorship : { $avg: "$iAnnual_Value" } } }, ],
...   "Sponsor_Count": [ { $group: { _id: "$Sponsor.Name", Sponsor_Count: { $sum: 1 } } }, ] } }
...
[
{
  Avg_Sponsorship: [
    { _id: 'Engineering', Avg_Sponsorship: 12300.970873786408 },
    { _id: 'Medicine', Avg_Sponsorship: 12113.207547169812 },
    { _id: 'Arts', Avg_Sponsorship: 12101.0101010101 },
    { _id: 'Business', Avg_Sponsorship: 12094.33962264151 },
    { _id: 'Science', Avg_Sponsorship: 12200 }
  ],
  Sponsor_Count: [
    { _id: 'Glen Dimplex', Sponsor_Count: 170 },
    { _id: 'Smurfit Kappa', Sponsor_Count: 175 },
    { _id: 'Greencore', Sponsor_Count: 174 }
  ]
}
]
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>
```

Screenshot 5.1

In this query, we applied \$match to filter manufacturing sector scholarships, \$unwind to expand the Discipline array, and \$facet to create 2 aggregation pipelines. We applied \$group with \$avg for average sponsorship per discipline and \$sum for scholarship count per person.

Aggregation Query 6

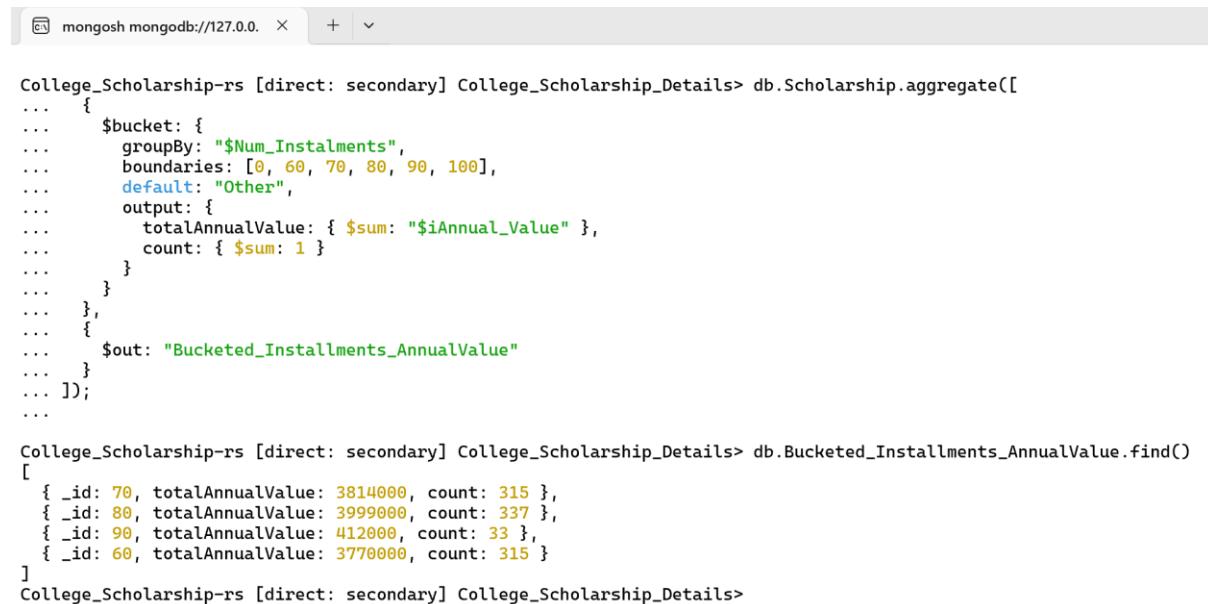
Question: How are scholarships distributed based on the **number of installments**, and what is the **total annual value** provided within each installment range?

Aggregate Query

```
db.Scholarship.aggregate([
  {
    $bucket: {
      groupBy: "$Num_Instalments",
      boundaries: [0, 60, 70, 80, 90, 100],
      default: "Other",
      output: {
        totalAnnualValue: { $sum: "$iAnnual_Value" },
        count: { $sum: 1 }
      }
    }
  },
  {
    $out: "Bucketed_Installments_AnnualValue"
  }
]);
```

Out Query

```
db.Bucketed_Installments_AnnualValue.find()
```



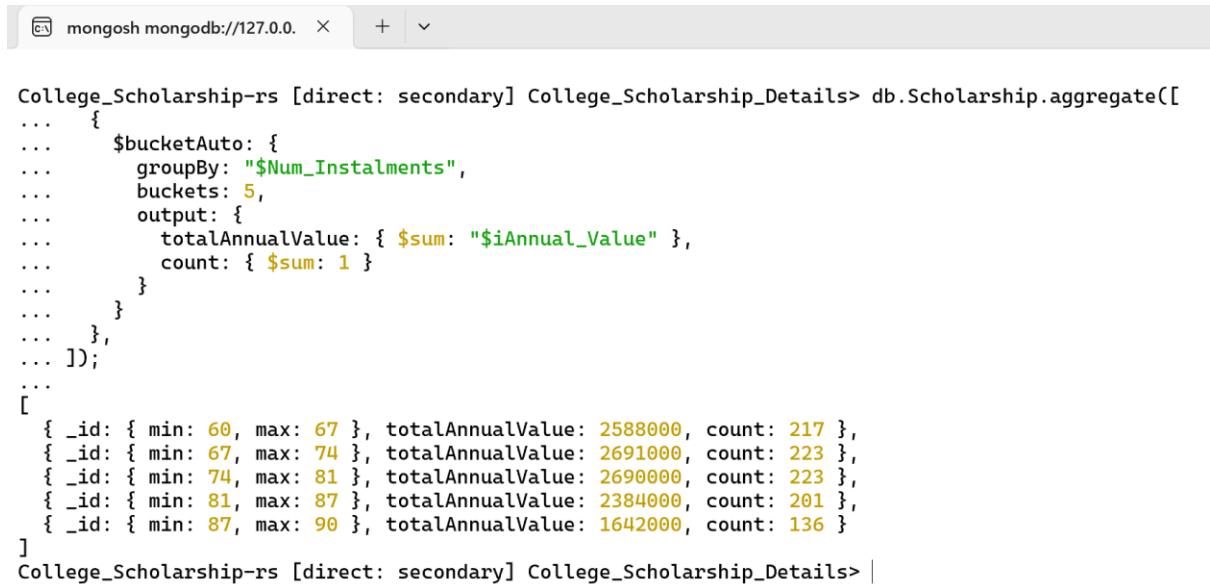
The screenshot shows a MongoDB shell session with the following content:

```
mongosh mongodb://127.0.0.1:27017
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.aggregate([
  ...
  {
    $bucket: {
      groupBy: "$Num_Instalments",
      boundaries: [0, 60, 70, 80, 90, 100],
      default: "Other",
      output: {
        totalAnnualValue: { $sum: "$iAnnual_Value" },
        count: { $sum: 1 }
      }
    }
  },
  {
    $out: "Bucketed_Installments_AnnualValue"
  }
]);
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Bucketed_Installments_AnnualValue.find()
[
  { _id: 70, totalAnnualValue: 3814000, count: 315 },
  { _id: 80, totalAnnualValue: 3999000, count: 337 },
  { _id: 90, totalAnnualValue: 4120000, count: 33 },
  { _id: 60, totalAnnualValue: 3770000, count: 315 }
]
College_Scholarship-rs [direct: secondary] College_Scholarship_Details>
```

Screenshot 6.1

Aggregate Query 6 with BucketAuto

```
db.Scholarship.aggregate([
  {
    $bucketAuto: {
      groupBy: "$Num_Instalments",
      buckets: 5,
      output: {
        totalAnnualValue: { $sum: "$iAnnual_Value" },
        count: { $sum: 1 }
      }
    }
  },
]);
```



The screenshot shows a MongoDB shell session titled "mongosh mongodb://127.0.0.1:27017". The command entered is an aggregate query targeting the "College_Scholarship_Details" collection. The query uses the \$bucketAuto stage to group documents by their "Num_Instalments" field into 5 automatic ranges. It then calculates the total annual value and the count of scholarships for each range. The results are displayed as an array of documents, each containing an "_id" field representing a range and fields for "totalAnnualValue" and "count".

```
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.aggregate([
...   {
...     $bucketAuto: {
...       groupBy: "$Num_Instalments",
...       buckets: 5,
...       output: {
...         totalAnnualValue: { $sum: "$iAnnual_Value" },
...         count: { $sum: 1 }
...       }
...     }
...   },
... ],
... );
[{"_id": {"min: 60, max: 67"}, "totalAnnualValue: 2588000, count: 217}, {"_id": {"min: 67, max: 74"}, "totalAnnualValue: 2691000, count: 223}, {"_id": {"min: 74, max: 81"}, "totalAnnualValue: 2690000, count: 223}, {"_id": {"min: 81, max: 87"}, "totalAnnualValue: 2384000, count: 201}, {"_id": {"min: 87, max: 90"}, "totalAnnualValue: 1642000, count: 136}]

College_Scholarship-rs [direct: secondary] College_Scholarship_Details> |
```

Screenshot 6.2

In this query, we used \$bucket with fixed boundaries in range of (0,60, 70, 80, 90, 100) and another query using \$bucketAuto for automatic ranges to group by installment ranges, and \$out to store the results from query. We applied \$sum to compute annual total value and scholarship count.

Aggregation Query 7

Question: Display the sponsor name, discipline count, college count, and total annual value for **top 5** sponsors with **Initial Payments** atleast **500**, The query should group the scholarships by **Sponsor Name**, count the unique **disciplines** and **colleges** they support, calculate the **annual total value** and filter out sponsor offering scholarships in atleast 3 **discipline** and 2 **colleges**. Results should be sorted in number of **discipline** and **total annual value** in descending order.

Aggregate Query

```
db.Scholarship.aggregate([
  { $match: { Initial_Payment_Num: { $gte: 500 } } },
  { $unwind: "$Discipline" },
  { $group: {
    _id: "$Sponsor.Name",
    disciplines: { $addToSet: "$Discipline" },
    colleges: { $addToSet: "$College.Name" },
    totalAnnualValue: { $sum: "$iAnnual_Value" },
    scholarshipCount: { $sum: 1 }
  } },
  { $addFields: {
    disciplineCount: { $size: "$disciplines" },
    collegeCount: { $size: "$colleges" }
  } },
  { $match: {
    disciplineCount: { $gte: 3 },
    collegeCount: { $gte: 2 }
  } },
  { $sort: {
    disciplineCount: -1,
    totalAnnualValue: -1
  } },
  { $limit: 5 },
  { $project: {
    _id: 0,
    sponsor: "$_id",
    disciplineCount: 1,
    collegeCount: 1,
    totalAnnualValue: 1
  } }
])
```

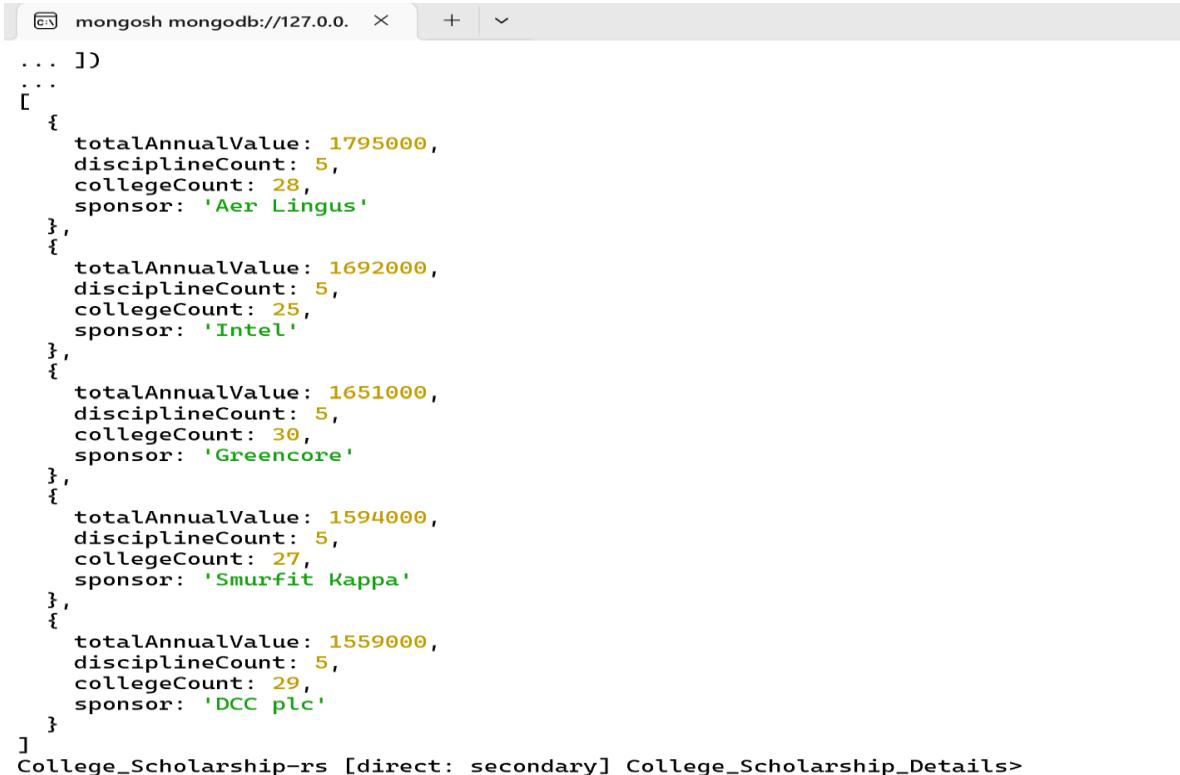


```

mongosh mongodb://127.0.0.1:27017/College_Scholarship_Details> db.Scholarship.aggregate([
...   { $match: { Initial_Payment_Num: { $gte: 500 } } },
...   { $unwind: "$Discipline" },
...   { $group: {
...     _id: "$Sponsor.Name",
...     disciplines: { $addToSet: "$Discipline" },
...     colleges: { $addToSet: "$College.Name" },
...     totalAnnualValue: { $sum: "$iAnnual_Value" },
...     scholarshipCount: { $sum: 1 }
...   } },
...   { $addFields: {
...     disciplineCount: { $size: "$disciplines" },
...     collegeCount: { $size: "$colleges" }
...   } },
...   { $match: {
...     disciplineCount: { $gte: 3 },
...     collegeCount: { $gte: 2 }
...   } },
...   { $sort: {
...     disciplineCount: -1,
...     totalAnnualValue: -1
...   } },
...   { $limit: 5 },
...   { $project: {
...     _id: 0,
...     sponsor: "$_id",
...     disciplineCount: 1,
...     collegeCount: 1,
...     totalAnnualValue: 1
...   } }
... ])
[{"sponsor": "Aer Lingus", "totalAnnualValue": 1795000, "collegeCount": 28, "disciplineCount": 5}, {"sponsor": "Intel", "totalAnnualValue": 1692000, "collegeCount": 25, "disciplineCount": 5}, {"sponsor": "Greencore", "totalAnnualValue": 1651000, "collegeCount": 30, "disciplineCount": 5}, {"sponsor": "Smurfit Kappa", "totalAnnualValue": 1594000, "collegeCount": 27, "disciplineCount": 5}, {"sponsor": "DCC plc", "totalAnnualValue": 1559000, "collegeCount": 29, "disciplineCount": 5}]

```

Screenshot 7.1



```

mongosh mongodb://127.0.0.1:27017/College_Scholarship_Details>

```

Screenshot 7.2

The pipeline filters scholarships with initial payments of atleast 500 unwinds the Discipline array to process each discipline, groups by sponsor to collect unique disciplines and colleges, calculates the total annual value, and counts scholarships. It then adds fields to count unique disciplines and colleges, filters for sponsors meeting diversity thresholds, sorts by discipline count and total annual value, limits to 5 results, and formats the output.

Aggregation Query 8

Question: Which five sponsors have provided the highest total scholarship value for scholarships that have **conditions** like Average mark above 70 or Must pass exams, and what are their **total scholarship value, average annual amount, and scholarship count?**

Aggregate Query

```
db.Scholarship.aggregate([
  { $match: { Conditions: { $in: ["Average mark above 70", "Must pass exams"] } } },
  { $addFields: { Total_Value: { $multiply: ["$iAnnual_Value", "$Duration"] } } },
  { $group: { _id: "$Sponsor.Name",
    Total_Scholarship_Value: { $sum: "$Total_Value" },
    Avg_Annual_Value: { $avg: "$iAnnual_Value" },
    Count: { $sum: 1 }
  } },
  { $sort: { Total_Scholarship_Value: -1 } },
  { $project: {
    _id: 0,
    Sponsor_Name: "$_id",
    Total_Scholarship_Value: 1,
    Avg_Annual_Value: { $round: ["$Avg_Annual_Value", 2] },
    Count: 1
  } },
  { $limit: 5
])
```



The screenshot shows the MongoDB shell interface with the command and its output. The command is an aggregation pipeline to find the top 5 sponsors based on total scholarship value, including their names, total values, average annual amounts, and counts. The output lists five sponsors: Aer Lingus, Intel, Greencore, Eir, and Ryanair, each with their respective statistics.

```
mongosh mongodb://127.0.0.1:27017/College_Scholarship_rs [direct: secondary] College_Scholarship_Details> db.Scholarship.aggregate([
...   { $match: { Conditions: { $in: ["Average mark above 70", "Must pass exams"] } } },
...   { $addFields: { Total_Value: { $multiply: ["$iAnnual_Value", "$Duration"] } } },
...   { $group: { _id: "$Sponsor.Name",
    ...     Total_Scholarship_Value: { $sum: "$Total_Value" },
    ...     Avg_Annual_Value: { $avg: "$iAnnual_Value" },
    ...     Count: { $sum: 1 }
  } },
...   { $sort: { Total_Scholarship_Value: -1 } },
...   { $limit: 5
...
]
{
  "_id": "Aer Lingus",
  "Total_Scholarship_Value": 2421000,
  "Avg_Annual_Value": 12303.57142857143,
  "Count": 56
},
{
  "_id": "Intel",
  "Total_Scholarship_Value": 2303000,
  "Avg_Annual_Value": 11888.888888888889,
  "Count": 54
},
{
  "_id": "Greencore",
  "Total_Scholarship_Value": 2236000,
  "Avg_Annual_Value": 12760,
  "Count": 50
},
{
  "_id": "Eir",
  "Total_Scholarship_Value": 2042000,
  "Avg_Annual_Value": 12086.95652173913,
  "Count": 46
},
{
  "_id": "Ryanair",
  "Total_Scholarship_Value": 2003000,
  "Avg_Annual_Value": 12184.166666666666,
  "Count": 48
}
]
```

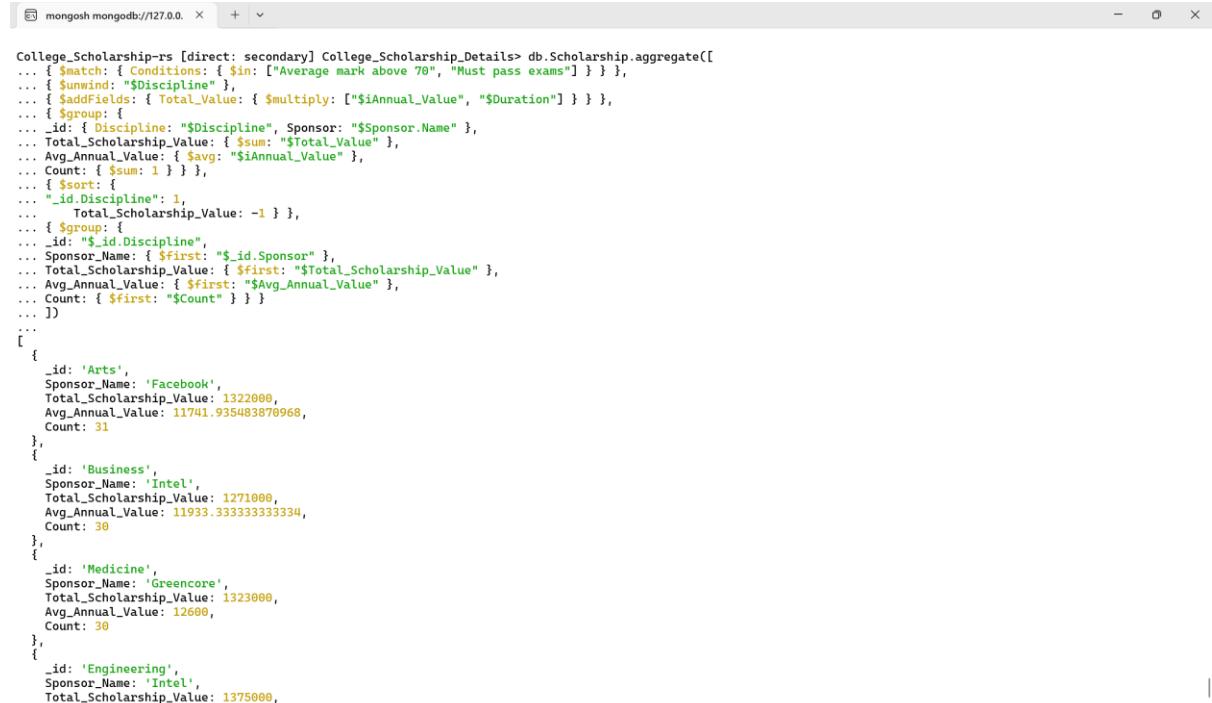
Screenshot 8.1

Aggregation Query 9

Question: Display **sponsors name** that have provided the highest total scholarship value for every **discipline** that have **conditions** like Average mark above 70 or Must pass exams, and what are their **total scholarship value, average annual amount, and scholarship count?**

Aggregate Query

```
db.Scholarship.aggregate([
  { $match: { Conditions: { $in: ["Average mark above 70", "Must pass exams"] } } },
  { $unwind: "$Discipline" },
  { $addFields: { Total_Value: { $multiply: ["$iAnnual_Value", "$Duration"] } } },
  { $group: {
    _id: { Discipline: "$Discipline", Sponsor: "$Sponsor.Name" },
    Total_Scholarship_Value: { $sum: "$Total_Value" },
    Avg_Annual_Value: { $avg: "$iAnnual_Value" },
    Count: { $sum: 1 } } },
  { $sort: {
    "_id.Discipline": 1,
    Total_Scholarship_Value: -1 } },
  { $group: {
    _id: "$_id.Discipline",
    Sponsor_Name: { $first: "$_id.Sponsor" },
    Total_Scholarship_Value: { $first: "$Total_Scholarship_Value" },
    Avg_Annual_Value: { $first: "$Avg_Annual_Value" },
    Count: { $first: "Count" } } }
])
```



The screenshot shows the MongoDB shell interface with the command `db.Scholarship.aggregate([...])` entered and its results displayed below. The results are grouped by discipline, showing the sponsor name, total scholarship value, average annual value, and count for each discipline.

```
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> db.Scholarship.aggregate([
... { $match: { Conditions: { $in: ["Average mark above 70", "Must pass exams"] } } },
... { $unwind: "$Discipline" },
... { $addFields: { Total_Value: { $multiply: ["$iAnnual_Value", "$Duration"] } } },
... { $group: {
...   _id: { Discipline: "$Discipline", Sponsor: "$Sponsor.Name" },
...   Total_Scholarship_Value: { $sum: "$Total_Value" },
...   Avg_Annual_Value: { $avg: "$iAnnual_Value" },
...   Count: { $sum: 1 } } },
... { $sort: {
...   "_id.Discipline": 1,
...   Total_Scholarship_Value: -1 } },
... { $group: {
...   _id: "$_id.Discipline",
...   Sponsor_Name: { $first: "$_id.Sponsor" },
...   Total_Scholarship_Value: { $first: "$Total_Scholarship_Value" },
...   Avg_Annual_Value: { $first: "$Avg_Annual_Value" },
...   Count: { $first: "Count" } } }
])
[{
  _id: 'Arts',
  Sponsor_Name: 'Facebook',
  Total_Scholarship_Value: 1322000,
  Avg_Annual_Value: 11741.935483870968,
  Count: 31
},
{
  _id: 'Business',
  Sponsor_Name: 'Intel',
  Total_Scholarship_Value: 1271000,
  Avg_Annual_Value: 11933.333333333334,
  Count: 30
},
{
  _id: 'Medicine',
  Sponsor_Name: 'Greencore',
  Total_Scholarship_Value: 1323000,
  Avg_Annual_Value: 12600,
  Count: 30
},
{
  _id: 'Engineering',
  Sponsor_Name: 'Intel',
  Total_Scholarship_Value: 1375000,
```

Screenshot 9.1

```
mongosh mongodb://127.0.0.1:27017/College_Scholarship_rs [direct: secondary] College_Scholarship_Details>
```

```
},
{
  _id: 'Engineering',
  Sponsor_Name: 'Intel',
  Total_Scholarship_Value: 13750000,
  Avg_Annual_Value: 11666.66666666666,
  Count: 33
},
{
  _id: 'Science',
  Sponsor_Name: 'Intel',
  Total_Scholarship_Value: 12000000,
  Avg_Annual_Value: 11413.793103448275,
  Count: 29
}
]
```

Screenshot 9.2

Section 4: Atlas Overview and Simple Queries

For the Atlas signed up for a MongoDB Atlas account, After logging into the account, Created new project called Scholarships and proceed with further set-up for a cluster by selecting cloud provider and region. Once the cluster was created a database user was created and access were allowed to that user.

Before importing the data directly from the existing dataset, I created a database directly into MongoDB Atlas in its built-in UI. By the use of Atlas query as shown in *Figure 1.1* to manually insert collections into newly created database. In the below *Figure 1.2* it shows completion of the insert document query.

The screenshot shows the MongoDB Atlas interface. On the left, there's a sidebar with various options like Cluster, Overview, Data Explorer (which is selected), Real Time, Cluster Metrics, Query Insights, Performance Advisor, Command Line Tools, and Infrastructure as Code. The main area shows the 'Organization' (Parimal's Org - 2025-...), 'Project' (College_Scholarships), and 'Cluster' (Scholarships). A modal window titled 'Insert Document' is open, showing a large JSON array of 1000 documents. The array starts with document 972 and ends at 1000. Each document has fields such as '_id', 'Title', 'Discipline', 'Breakdown', 'Annual_Value', 'Duration', 'Initial_Payment_Num', 'Num_Instalments', 'Category', 'Conditions', 'College', and 'Sponsor'. The total size of the data is listed as 44KB. At the bottom of the modal, there are 'Cancel' and 'Insert' buttons.

Figure 1.1

The screenshot shows the MongoDB Atlas interface again. The left sidebar is identical to Figure 1.1. The main area now shows the 'College_Scholarships' collection under the 'Scholarships' section. A modal window titled 'College_Scholarships.Scholarships' is open, displaying the results of the inserted documents. The results are filtered to show only 1-5 of 5. The first result is for 'HopeSchool' with ID 24, and the second is for 'InspireU' with ID 392. Both results show the same fields as the inserted documents in Figure 1.1. At the bottom of the modal, there are 'Reset', 'Apply', and 'Options' buttons.

Figure 1.2

After manually setting up the database, I explored to the MongoDB Atlas cluster using the shell by copying the connection from local replica set into Atlas system. Used the mongorestore command to import the database from local replica set into MongoDB Atlas Environment.

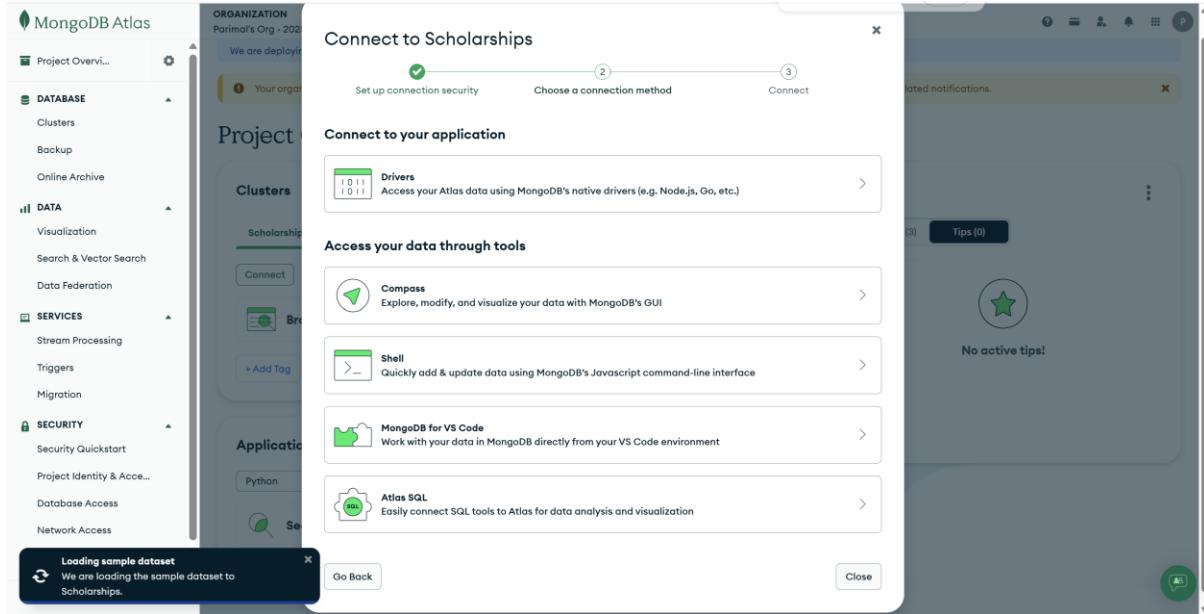


Figure 1.3

The command below creates a backup of the existing database from local machine using authentication and saves into a folder called dump.

mongodump –uri =

```
"mongodb://system:system@localhost:27017,localhost:27018,localhost:27019/?replicaSet=College_Scholarship-rs&authSource=admin" --db=College_Scholarship_Details --out=dump/
```

This command restores the locally dumped dataset from the Local Replica set into Scholarships database into Mongodb Atlas environment.

mongorestore –uri =

```
"mongodump+srv://system:system@scholarships.zcywcw6.mongodb.net/Scholarships" --dir=dump/College_Scholarship_Details --drop
```

```
C:\Users\parim>mongodump --uri="mongodb://system:system@localhost:27017,localhost:27018,localhost:27019/?replicaSet=College_Scholarship-rs&authSource=admin" --db=College_Scholarship_Details --out=dump/
2025-04-29T15:48:45.587+0100      writing College_Scholarship_Details.Bucketed_Installments_AnnualValue to dump\College_Scholarship_Details\Bucketed_Installments_AnnualValue.bson
2025-04-29T15:48:45.591+0100      done dumping College_Scholarship_Details.Bucketed_Installments_AnnualValue (4 documents)
2025-04-29T15:48:45.604+0100      writing College_Scholarship_Details.Scholarship to dump\College_Scholarship_Details\Scholarship.bson
2025-04-29T15:48:45.609+0100      done dumping College_Scholarship_Details.Scholarship (1000 documents)

C:\Users\parim>mongorestore --uri="mongodb+srv://system:system@scholarships.zcywcw6.mongodb.net/Scholarships" --dir=dump/College_Scholarship_Details --drop
2025-04-29T15:48:57.413+0100      The --db and --collection flags are deprecated for this use-case; please use --nsInclude instead, i.e. with --nsInclud
ude=${DATABASE}.${COLLECTION}
2025-04-29T15:48:57.414+0100      building a list of collections to restore from dump\College_Scholarship_Details dir
2025-04-29T15:48:57.414+0100      don't know what to do with file "dump\College_Scholarship_Details\prelude.json", skipping...
2025-04-29T15:48:57.415+0100      reading metadata for Scholarships.Bucketed_Installments_AnnualValue from dump\College_Scholarship_Details\Bucketed_Instal
lments_AnnualValue.metadata.json
2025-04-29T15:48:57.415+0100      reading metadata for Scholarships.Scholarship from dump\College_Scholarship_Details\Scholarship.metadata.json
2025-04-29T15:48:57.475+0100      restoring Scholarships.Scholarship from dump\College_Scholarship_Details\Scholarship.bson
2025-04-29T15:48:57.503+0100      restoring Scholarships.Bucketed_Installments_AnnualValue from dump\College_Scholarship_Details\Bucketed_Installments_AnnualValue.bson
2025-04-29T15:48:57.626+0100      finished restoring Scholarships.Bucketed_Installments_AnnualValue (4 documents, 0 failures)
2025-04-29T15:49:00.217+0100      finished restoring Scholarships.Scholarship (1000 documents, 0 failures)
2025-04-29T15:49:00.217+0100      no indexes to restore for collection Scholarships.Scholarship
2025-04-29T15:49:00.217+0100      no indexes to restore for collection Scholarships.Bucketed_Installments_AnnualValue
2025-04-29T15:49:00.217+0100      1004 document(s) restored successfully. 0 document(s) failed to restore.

C:\Users\parim>
```

Figure 1.3

As in *Figure 1.4* the restore operation was successful where both the collections Scholarship and Bucketed_Installment_AnnualValue were restored and now visible in the Atlas database.

The screenshot shows the MongoDB Compass interface. The left sidebar has 'Cluster' selected, with 'Data Explorer' highlighted. The main area shows the 'Data' tab for the 'Scholarships' collection. It displays storage details: 120KB storage size, 481.3KB logical data size, 1000 total documents, and 44KB index size. There are tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation', and 'Search Indexes'. A search bar says 'Generate queries from natural language in Compass®'. Below is a 'Filter' section with a query input: 'Type a query: { field: 'value' }'. An 'INSERT DOCUMENT' button is at the top right. The bottom shows a list of 1-20 results, with the first document partially expanded:

```
_id: 16
  Title: "STEM Boost"
  Discipline: Array (2)
  Branches: Array (2)
  {Annual_Value : 15000}
  Duration : 4
  Initial_Payment_Num : 1000
  Num_Instalments : 66
  Category : "Postgraduate"
  Conditions : "Average mark above 70"
  College : Object
  Sponsor : Object
```

Figure 1.4

I navigated through the Data Explorer Tab in MongoDB atlas and selected the scholarships collection within my cluster. I explored using the Find feature which provides easy and a smooth user interface to view and filter out the data, it also provides easy modification, updation and deletion in the collection.

Simple Query 1 in Atlas – Simple Query 5

```
$nor: [{ "Sponsor.Sector": "Banking" }, { "Sponsor.Sector": "Airlines" }],
        Discipline: "Engineering",
        "College.Name": "Athlone IT",
        iAnnual_Value: { $gte: 12000 }
```

The screenshot shows the MongoDB Atlas interface. On the left, there's a sidebar with options like Cluster, Overview, Data Explorer (which is selected and highlighted in green), RediTime, Cluster Metrics, Query Insights, Performance Advisor, Command Line Tools, and Infrastructure as Code. The main area has tabs for ORGANIZATION, PROJECT, and CLUSTER, with 'College_Scholarships' selected under CLUSTER. In the center, there's a search bar and a sidebar for 'College_Scholarships.Scholarships' with 'Scholarships' selected. Below that is a 'Filter' section containing the query code provided above. To the right of the filter is a 'QUERY RESULTS: 1-4 OF 4' section showing four document snippets. At the bottom, there's a footer with 'System Status: All Good' and links to Status, Terms, Privacy, Atlas Blog, and Contact Sales.

Figure 2.1

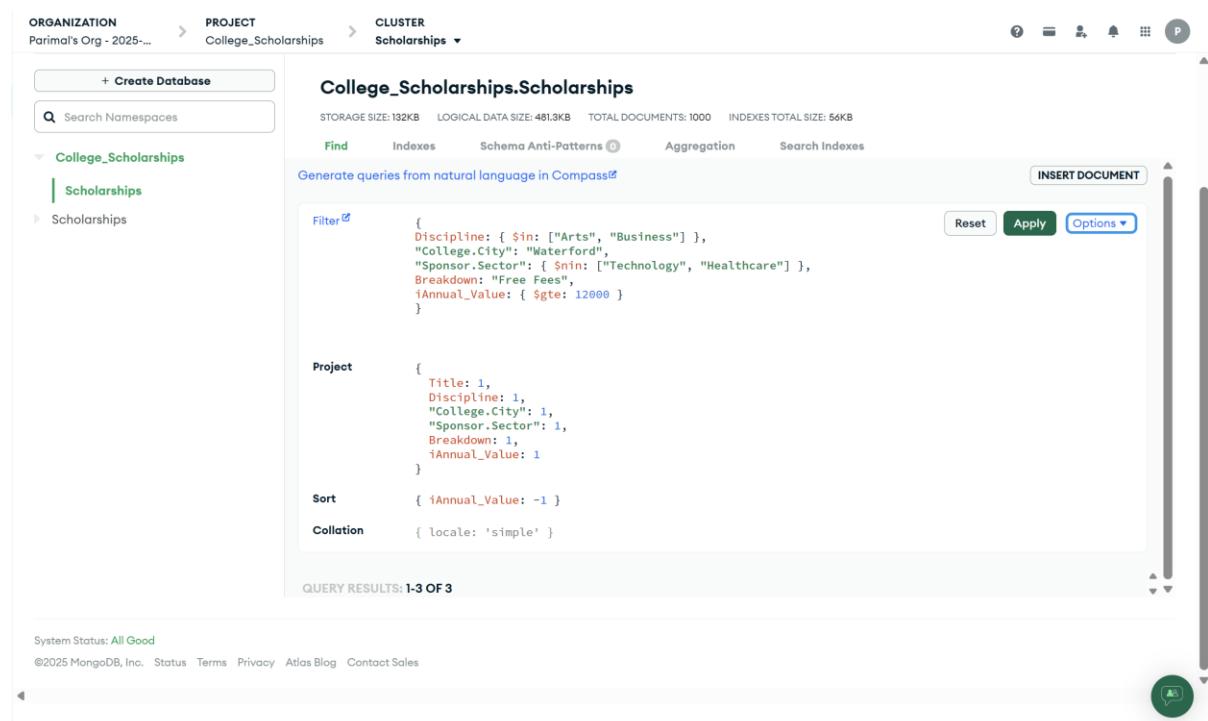
In the above simple query, I have replicated the same simple query 5 executed in the MongoDB shell, where the sponsor sector was either "Banking" or "Airlines", and matched additional criteria such as discipline being "Engineering", the college name as "Athlone IT", and the annual value being greater than or equal to 12,000. The query returned 4 documents, demonstrating how Atlas could be direct and efficient in data querying.

Simple Query 2 in Atlas – Simple Query 17

Filter: { **Discipline**: { \$in: ["Arts", "Business"] },
"College.City": "Waterford",
"Sponsor.Sector": { \$nin: ["Technology", "Healthcare"] },
Breakdown: "Free Fees",
iAnnual_Value: { \$gte: 12000 } }

Project: { **Title**: 1,
 Discipline: 1,
 "College.City": 1,
 "Sponsor.Sector": 1,
 Breakdown: 1,
 iAnnual_Value: 1 }

Sort: { **iAnnual_Value**: -1 }



The screenshot shows the MongoDB Compass interface. The top navigation bar indicates the database is 'Parimal's Org - 2025...', the project is 'College_Scholarships', and the cluster is 'Scholarships'. The main area displays a query for the 'College_Scholarships.Scholarships' collection. The query details are as follows:

```
Filter: { Discipline: { $in: ["Arts", "Business"] }, "College.City": "Waterford", "Sponsor.Sector": { $nin: ["Technology", "Healthcare"] }, Breakdown: "Free Fees", iAnnual_Value: { $gte: 12000 } }
```

The 'Project' section shows the projection for the results:

```
Project: { Title: 1, Discipline: 1, "College.City": 1, "Sponsor.Sector": 1, Breakdown: 1, iAnnual_Value: 1 }
```

The 'Sort' section shows the sorting criteria:

```
Sort: { iAnnual_Value: -1 }
```

The 'Collation' section shows the collation setting:

```
Collation: { locale: 'simple' }
```

At the bottom, it says 'QUERY RESULTS: 1-3 OF 3'.

Figure 2.2

DATABASES: 2 COLLECTIONS: 3

College_Scholarships.Scholarships

STORAGE SIZE: 132KB LOCAL DATA SIZE: 481.3KB TOTAL DOCUMENTS: 1000 INDEXES TOTAL SIZE: 56KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

Generate queries from natural language in Compass

Filter

```
{
  "Discipline": { $in: ["Arts", "Business"] },
  "College.City": "Waterford",
  "Sponsor.Sector": { $nin: ["Technology", "Healthcare"] },
  "Breakdown": "Free Fees",
  "Annual_Value": { $gte: 12000 }
}
```

QUERY RESULTS: 1-3 OF 3

```

_id: 603
Title: "Code4All"
Discipline: Array (4)
Breakdown: Array (2)
Annual_Value: 14000
College: Object
Sponsor: Object

_id: 210
Title: "GreenFdu"
Discipline: Array (4)
Breakdown: Array (3)
Annual_Value: 12000
College: Object
Sponsor: Object

```

System Status: All Good
©2025 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

Figure 2.3

In the above query is replicated from simple query 17, I continued the exploration inside find feature in MongoDB Atlas, which supports various query types such as filters, projection, and sorting. A query was run to retrieve scholarships from City called "Waterford" where the discipline was in either "Arts" or "Business", and the sponsor sector did not include both "Technology" and "Healthcare". Additionally, filtration was applied that included "Free Fees" and had an annual value of at least 12,000. I applied a projection to display specific fields and used sorting to arrange the results by Annual Value in descending annual order. This helped me understand how different **Find** options work together to refine data retrieval.

Section 5: Atlas Aggregation Framework

The screenshot shows the MongoDB Atlas Aggregation Framework interface. The pipeline stage is set to '\$match'. The output after the '\$match' stage (sample of 10 documents) is displayed. One document example is shown:

```

1 {
  "Sponsor.Sector": "Manufacturing",
}

```

The preview pane shows the original document structure and the modified document after the '\$match' stage.

Figure 3.1 – Aggregation Stage 1

The screenshot shows the MongoDB Atlas Aggregation Framework interface. The pipeline stages are '\$match', '\$unwind', and '\$facet'. The output after the '\$unwind' stage (sample of 10 documents) is displayed. One document example is shown:

```

1 {
  "$Discipline": "Engineering"
}

```

The preview pane shows the original document structure and the modified document after the '\$unwind' stage. The output after the '\$facet' stage (sample of 1 document) is also shown.

Figure 3.2 – Aggregation Stage 2

The screenshot shows the MongoDB Atlas aggregation pipeline editor. The top navigation bar displays 'ORGANIZATION' (Parimal's Org - 2025...), 'PROJECT' (College_Scholarships), and 'CLUSTER' (Scholarships). Below the navigation, it shows 'DATABASES: 2' and 'COLLECTIONS: 3'. On the right, there are buttons for 'VISUALIZE YOUR DATA' and 'REFRESH'.

The main interface is titled 'College_Scholarships.Scholarships' and shows document statistics: 'STORAGE SIZE: 132KB', 'LOGICAL DATA SIZE: 481.3KB', 'TOTAL DOCUMENTS: 1000', and 'INDEXES TOTAL SIZE: 56KB'. Below this are tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation' (which is selected), and 'Search Indexes'.

The 'Pipeline' section shows three stages: '\$match', '\$unwind', and '\$facet'. The '\$facet' stage is expanded, showing its code:

```

1 * {
2 *   Avg_Sponsorship: [
3 *     {
4 *       $group: {
5 *         _id: "$Discipline",
6 *         Avg_Sponsorship: {
7 *           $avg: "$Annual_Value",
8 *           },
9 *         },
10 *       ],
11 *     },
12 *     Sponsor_Count: [
13 *       {
14 *         $group: {
15 *           id: "$Sponsor.Name",
16 *           Sponsor_Count: {
17 *             $sum: 1,
18 *           },
19 *         },
20 *       ],
21 *     ],
22 *   }

```

To the right, the 'Output after \$facet stage (Sample of 1 document)' is displayed:

```

* Avg_Sponsorship: Array (5)
* 0: Object
  Avg_Sponsorship: 12300.979873786488
  _id: "Engineering"
* 1: Object
  Avg_Sponsorship: 12101.0101010101
  _id: "Arts"
* 2: Object
  Avg_Sponsorship: 12094.33962264151
  _id: "Business"
* 3: Object
  Avg_Sponsorship: 12200
  _id: "Science"
* 4: Object
  Avg_Sponsorship: 12113.207547169812
  _id: "Medicine"
* Sponsor_Count: Array (3)

```

At the bottom, there is a '+ Add Stage' button.

Figure 3.3 – Aggregation Stage 3

The aggregation pipeline in MongoDB Atlas was built using a stage to stage approach, which made the process easier to structure and visually follow.

Stage 1 - \$match: Filtered the dataset to include only scholarships by companies sector in Manufacturing.

Stage 2 - \$unwind: Expanded the Discipline array so that each discipline could be treated as an individual record.

Stage 3 - \$facet: Computed two insights, the average annual sponsorship value per discipline and the number of scholarships each sponsor provided.

See Figure 3.1 – 3.3 for visual snapshots for each stage.

Aggregate Query 2 in MongoDb Atlas – Replicated from Aggregate Query 4

```
[{ $addFields: {
    Total_Scholarship: { $multiply: ["$Num_Instalments", "$Initial_Payment_Num"] } },
  { $group: { _id: "$Sponsor.Sector",
    Total_Scholarship_Amount: { $sum: "$Total_Scholarship" },
    Scholarship_Count: { $sum: 1 } } },
  { $sort: { Total_Sponsor_Amount: -1 } } ]
```

The screenshot shows the MongoDB Atlas interface for managing databases and clusters. The current view is on the 'College_Scholarships' cluster under the 'Scholarships' collection. The pipeline editor is open, displaying the following aggregation pipeline:

```

1 * [
2 *   {
3 *     $addFields: {
4 *       Total_Scholarship: {
5 *         $multiply: [
6 *           "$Num_Instalments",
7 *           "$Initial_Payment_Num"
8 *         ],
9 *       },
10 *     },
11 *   },
12 *   {
13 *     $group: {
14 *       _id: "$Sponsor.Sector",
15 *       Total_Scholarship_Amount: { $sum: "$Total_Scholarship" },
16 *       Scholarship_Count: { $sum: 1 },
17 *     },
18 *   },
19 *   {
20 *     $sort: {
21 *       Total_Sponsor_Amount: -1,
22 *     },
23 *   }
24 * ]
25
26
27 ]
28

```

The 'PREVIEW' tab is selected, showing the pipeline output. The results are as follows:

- _id: "Banking"**
Total_Scholarship_Amount : 7888947
Scholarship_Count : 199
- _id: "Airlines"**
Total_Scholarship_Amount : 6204519
Scholarship_Count : 147
- Scholarship_Count : 73**
_id: "Construction"
Total_Scholarship_Amount : 3800427
- _id: "Manufacturing"**
Total_Scholarship_Amount : 8520544
Scholarship_Count : 208
- _id: "Technology"**

Figure 3.4 – Aggregation Query 2

In this query we have replicated the same query aggregate query 4 where, I used the command styled aggregate editor in MongoDb Atlas, which allowed the pipeline to automatically divide the logic into different stages. The query is replicated query of the one that is done in Aggregation Framework part which shows \$addField stage that calculates a new field, Total_Scholarships the number of instalments with the initial payment amount. \$group stage that summarises the total scholarship amount and count of scholarships by sponsor sector. Finally, a \$sort that sorts total sponsor amount in descending order. This command styled aggregate editor made the query both professional and easy to interpret.

Section 6: Atlas Dashboard

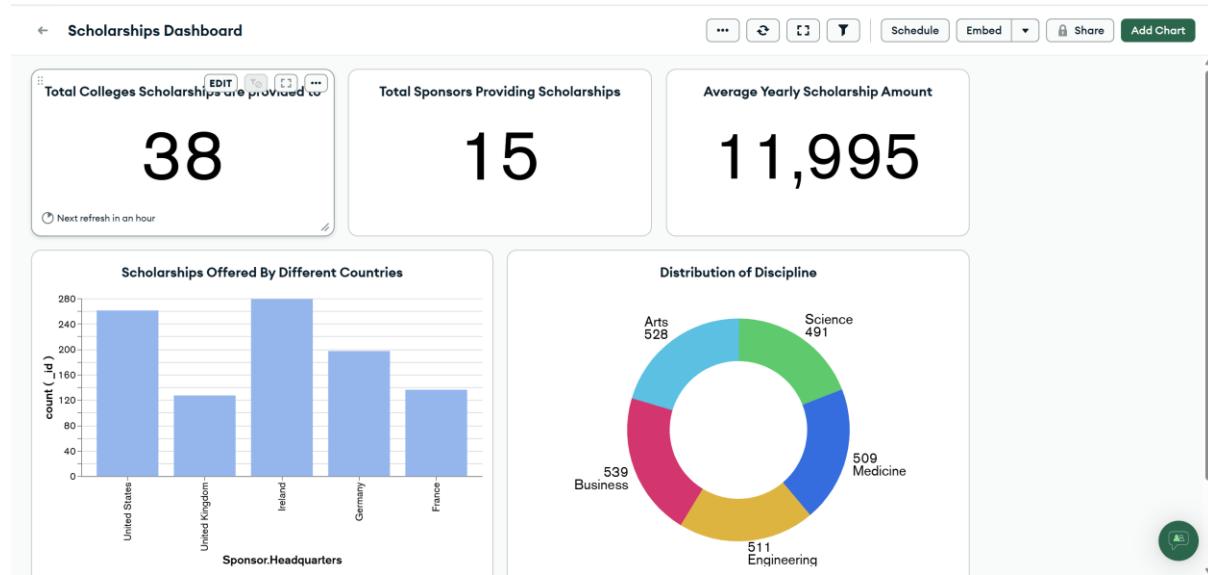


Figure 4.1

MongoDb Atlas offers a powerful visualisation feature which is not available in MongoDb compass version. It allows user to create powerful visualisation from the collection dataset. I explored this feature from the Data Visualisation tab and created custom Scholarship Dashboard, where I have added multiple charts including a bar chart, donut chart and multiple number fields to show basic insight about the collection. See Figure 4.1 for the dashboard view screenshot.

Issues Faced while using MongoDb Atlas

While working with the MongoDb Atlas, I encountered issue where the cluster failed to load and displayed an unexpected error. This occurred repeatedly because the replica set terminal window was closed, causing an connection break. Ensuring the cluster's association command prompt remains open the issue was then resolved. Refer Figure 5.1 for snip

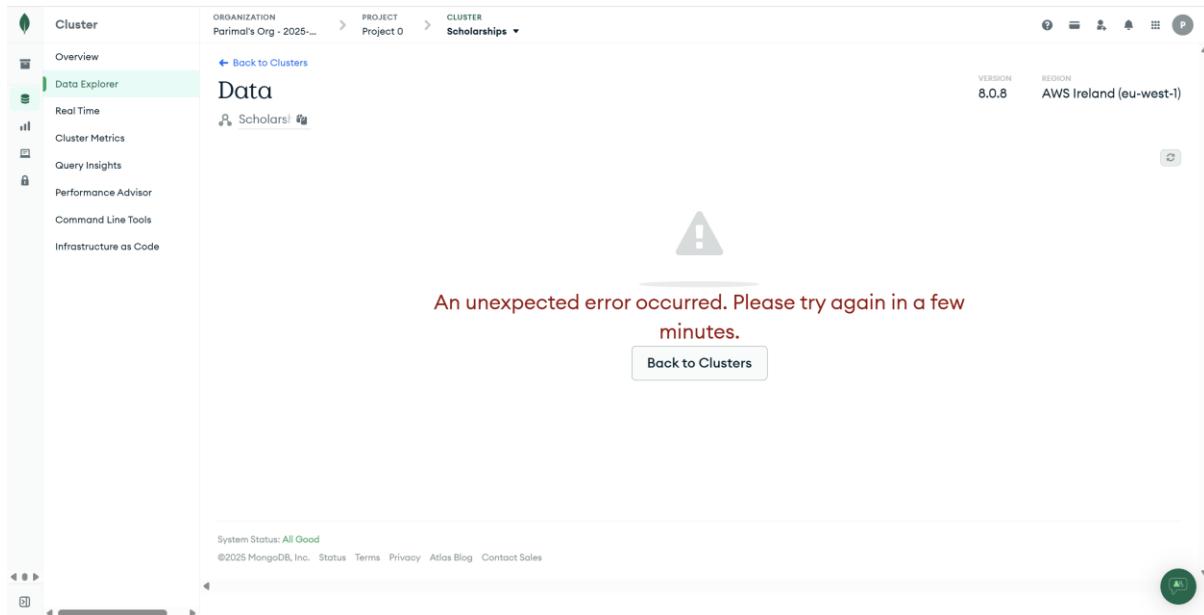


Figure 5.1

I encountered another challenge while connecting MongoDb Atlas to my local replica set during the data import process. Several attempts to use mongodump and mongorestore commands failed due to authentication errors, invalid URL's, and incorrect path for storing the dump file. This made it difficult to transfer the data from replica set to Atlas until the connection was correctly configured.

A screenshot of a Windows Command Prompt window. The title bar says 'Command Prompt'. The console output shows several attempts to run mongodump and mongorestore commands, all of which fail due to authentication issues or command line parsing errors. The user is trying to connect to a MongoDB instance at 'localhost:27017' and 'localhost:27018' using various URI formats and options like '--uri', '--db', and '--out'. The errors include 'Failed: can't create session', 'connection() error occurred during connection handshake', 'Authentication failed', 'error parsing command line options: error parsing uri: lookup <cluster>.mongodb.net: dnsquery: DNS name contains an invalid character', 'no such host', and 'The --db and --collection flags are deprecated for this use-case; please use --nsInclude instead, i.e. with --nsInclude \${DATABASE}.\${COLLECTION}'.

Figure 5.2

Section 7: Compass Overview and Queries

After exploring Atlas, with curiosity to check onto MongoDb Compass to explore the data locally. I tried configuring a connection using my local replica set, authenticated with credentials and successfully connected to manage data through Compass GUI. See Fig 6.1 for connection setup.

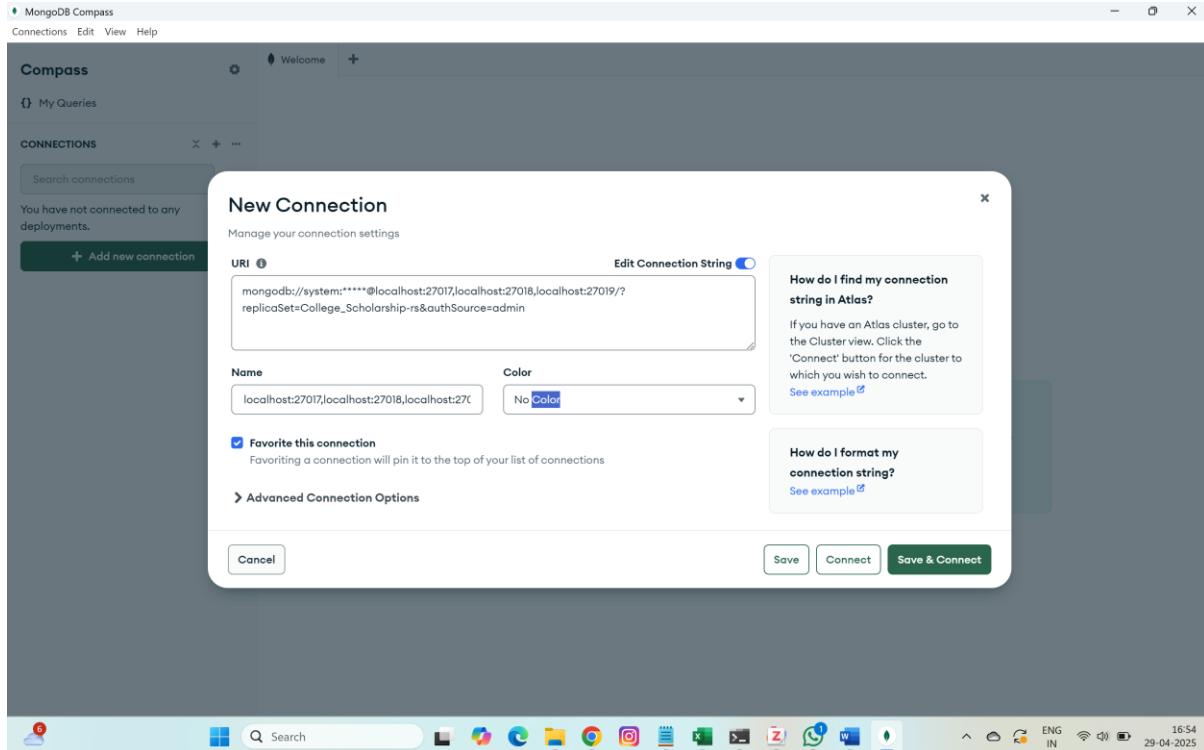


Figure 6.1

MongoDb Compass Importing Dataset from Local Replica set

Since I had already imported the dataset into MongoDb Atlas, importing the dataset into Compass was comparatively easier as I already knew the authentication and other processes. The local replica set was easily configured, allowing to explore the connections smoothly in Compass interface.

The screenshot shows the MongoDB Compass interface. The top navigation bar includes 'Connections', 'Edit', 'View', 'Collection', and 'Help'. Below the navigation is a header with tabs for 'Welcome', 'Scholarship', and a '+' button. The main area shows a 'CONNECTIONS' sidebar with a search bar and a list of connections: 'localhost:27017', 'localhost:27018', 'localhost:27019', 'College_Scholarship_Details', 'Bucketed_Installments_Annual...', and 'Scholarship'. The 'Scholarship' connection is selected. The main content area displays the 'Documents' tab with 1K documents. A search bar at the top says 'Type a query: { field: 'value' } or Generate query'. Below it are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. A toolbar with various icons is at the bottom right. The main pane shows three document snippets:

```
_id: 8
Title: "Code4All"
Discipline: Array (2)
Breakdown: Array (4)
iAnnual_Value: 10000
Duration: 3
Initial_Payment_Num: 345
Num_Instalments: 87
Category: "Postgraduate"
Conditions: "Must pass exams"
College: Object
Sponsor: Object

_id: 11
Title: "TechStart"
Discipline: Array (3)
Breakdown: Array (1)
iAnnual_Value: 12000
Duration: 3
Initial_Payment_Num: 450
Num_Instalments: 80
Category: "Postgraduate"
Conditions: "Average mark above 70"
College: Object
Sponsor: Object

_id: 43
Title: "GradFund"
Discipline: Array (2)
```

Figure 6.2

I tried running the same find query in the MongoDB Compass environment that I had previously executed in Atlas. The query filtered based on multiple conditions including Discipline, City, Sponsor.Sector, and Breakdown, and applied projection and sorting. While the structure and result of the query remained consistent, the runtime performance was noticeably more efficient in Compass, as it queried a local replica set, avoiding the network latency involved with cloud-based access. Additionally, Compass provided real-time visibility into the document structure, making it easier to validate embedded documents and array fields visually. This further reinforced how Compass can be highly effective for interactive local testing and debugging during development.

The screenshot shows the MongoDB Compass interface with the following details:

- Left Sidebar (Connections):** Shows a tree view of connections. The 'Scholarship' connection is selected, and its sub-collection 'Bucketed_Installments_Annual...' is also selected.
- Top Bar:** Displays the database path: localhost:27017,localhost:27018,localhost:27019 > College_Scholarship_Details > Scholarship.
- Toolbar:** Includes buttons for 'Generate query', 'Explain', 'Reset', 'Find', 'Options', 'EXPORT DATA', and 'INSIGHT'.
- Query Editor:** Shows the query text and its breakdown into Project, Sort, Collation, and Index Hint sections.
- Results Panel:** Displays two document results. The first document has an _id of 603, a Title of "Code4All", and an iAnnual_Value of 14000. The second document has an _id of 210, a Title of "GreenEdu".
- Bottom Navigation:** Includes a 'Max Time MS' input (set to 60000), 'Skip' (0), 'Limit' (0), and a page navigation bar showing 1-3 of 3.

Figure 6.3

This screenshot is identical to Figure 6.3, showing the same query results and interface elements for the 'Scholarship' collection in MongoDB Compass.

Figure 6.4

Section 8: Replica set

For setting up the Replica set, it was the most difficult process. Below is walkthrough for the whole process for Replica set.

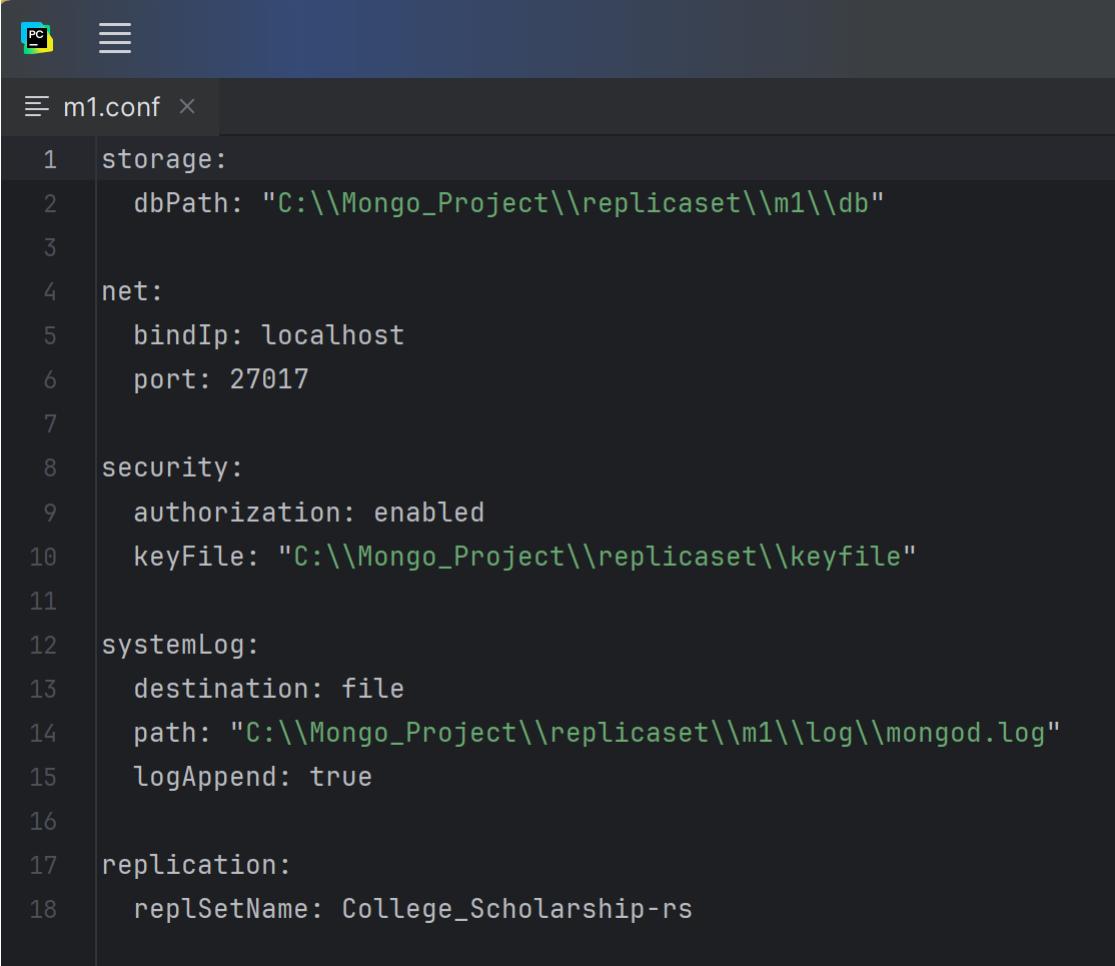
1. **Create Data Directories:** I set up 3 data directories for different instances:

m1\db
m2\db
m3\db

Name	Date modified	Type
m1	13-04-2025 03:03	File folder
m2	13-04-2025 03:03	File folder
m3	13-04-2025 03:03	File folder

Figure 7.1

2. **Create Configure Files:** To streamline and manage each node consistently, 3 separate .conf files were created for each instance. These files specify paths, ports, and replica set configurations.



```
storage:
  dbPath: "C:\\Mongo_Project\\replicaset\\m1\\db"

net:
  bindIp: localhost
  port: 27017

security:
  authorization: enabled
  keyFile: "C:\\Mongo_Project\\replicaset\\keyfile"

systemLog:
  destination: file
  path: "C:\\Mongo_Project\\replicaset\\m1\\log\\mongod.log"
  logAppend: true

replication:
  replSetName: College_Scholarship-rs
```

Figure 7.2

- 3. Start MongoDB Instances Using Config Files:** Each instance of .conf file was started in command prompt making every port enabled and working.

```
mongod --config C:\Mongo_Project\replicaset\m1\m1.conf
mongod --config C:\Mongo_Project\replicaset\m2\m2.conf
mongod --config C:\Mongo_Project\replicaset\m3\m3.conf
```

- 4. Initialise the dataset:** After the nodes are configured, connected to the first instance via shell and initialised the replica set.

```
mongosh --port 27017
```

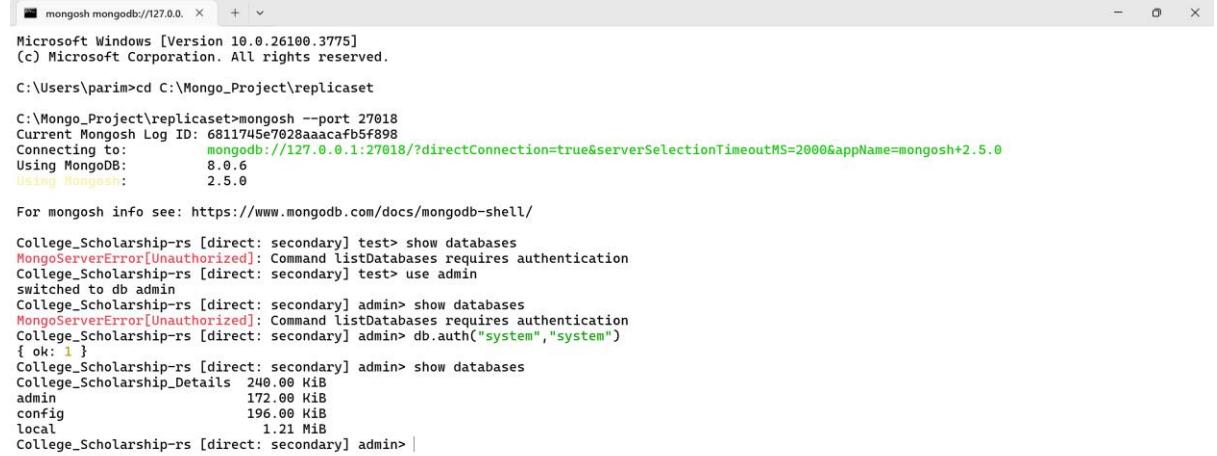
```
rs.initiate({
  _id: "College_Scholarship-rs",
  members: [
    { _id: 0, host: "localhost:27017" },
    { _id: 1, host: "localhost:27018" },
    { _id: 2, host: "localhost:27019" }
  ]
})
```

- 5. Verification:** The rs.status() confirms that the replica set is initialised and working correctly.

```
mongosh mongodb://127.0.0.1:27017
College_Scholarship-rs [direct: secondary] College_Scholarship_Details> rs.status()
{
  set: 'College_Scholarship-rs',
  date: ISODate('2025-04-30T00:58:49.091Z'),
  myState: 2,
  term: Long('123'),
  syncSourceHost: 'localhost:27019',
  syncSourceId: 2,
  heartbeatIntervalMillis: Long('2000'),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1745974249, i: 1 }), t: Long('123') },
    lastCommittedWallTime: ISODate('2025-04-30T00:58:49.057Z'),
    readConcernMajorityOptime: { ts: Timestamp({ t: 1745974249, i: 1 }), t: Long('123') },
    appliedOptime: { ts: Timestamp({ t: 1745974249, i: 1 }), t: Long('123') },
    durableOptime: { ts: Timestamp({ t: 1745974249, i: 1 }), t: Long('123') },
    writtenOptime: { ts: Timestamp({ t: 1745974249, i: 1 }), t: Long('123') },
    lastAppliedWallTime: ISODate('2025-04-30T00:58:49.057Z'),
    lastDurableWallTime: ISODate('2025-04-30T00:58:49.057Z'),
    lastWrittenWallTime: ISODate('2025-04-30T00:58:49.057Z')
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1745974219, i: 1 }),
  electionParticipantMetrics: {
    votedForCandidate: true,
    electionTerm: Long('123'),
    lastVoteDate: ISODate('2025-04-30T00:48:49.037Z'),
    electionCandidateMemberId: 2,
    voteReason: ''
  },
  lastWrittenOptimeAtElection: { ts: Timestamp({ t: 1745973787, i: 1 }), t: Long('122') },
  maxWrittenOptimeInSet: { ts: Timestamp({ t: 1745973787, i: 1 }), t: Long('122') },
  lastAppliedOptimeAtElection: { ts: Timestamp({ t: 1745973787, i: 1 }), t: Long('122') },
  maxAppliedOptimeInSet: { ts: Timestamp({ t: 1745973787, i: 1 }), t: Long('122') },
  priorityAtElection: 1,
  newTermStartDate: ISODate('2025-04-30T00:48:49.045Z'),
  newTermAppliedDate: ISODate('2025-04-30T00:48:49.062Z')
},
members: [
  {
    _id: 0,
    name: 'localhost:27017',
    health: 1,
    state: 2,
    stateStr: 'SECONDARY',
    uptime: 1306334
  }
]
```

Figure 7.3

In this step, I am connected to localhost port 27018 in Figure 7.5 and port 27019 in Figure 7.6. Although each instance is running independently, the same set of databases was accessible after authentication because of the shared dataset managed by replica set.



```

mongosh mongodb://127.0.0.1:27018
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\parim>cd C:\Mongo_Project\replicaset

C:\Mongo_Project\replicaset>mongosh --port 27018
Current Mongosh Log ID: 6811745e7028aaacaf5f898
Connecting to: mongodb://127.0.0.1:27018/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.0
Using MongoDB: 8.0.6
Using Mongosh: 2.5.0

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

College_Scholarship-rs [direct: secondary] test> show databases
MongoServerError[Unauthorized]: Command listDatabases requires authentication
College_Scholarship-rs [direct: secondary] test> use admin
switched to db admin
College_Scholarship-rs [direct: secondary] admin> show databases
MongoServerError[Unauthorized]: Command listDatabases requires authentication
College_Scholarship-rs [direct: secondary] admin> db.auth("system", "system")
{ ok: 1 }
College_Scholarship-rs [direct: secondary] admin> show databases
College_Scholarship_Details 240.00 KiB
admin 172.00 KiB
config 196.00 KiB
local 1.21 MiB
College_Scholarship-rs [direct: secondary] admin>

```

Figure 7.5



```

mongosh mongodb://127.0.0.1:27019
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\parim>mongosh --port 27019
Current Mongosh Log ID: 681174ce85e3376058b5f898
Connecting to: mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.0
Using MongoDB: 8.0.6
Using Mongosh: 2.5.0

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

College_Scholarship-rs [direct: primary] test> use admin
switched to db admin
College_Scholarship-rs [direct: primary] admin> db.auth("system", "system")
{ ok: 1 }
College_Scholarship-rs [direct: primary] admin> show databases
College_Scholarship_Details 200.00 KiB
admin 172.00 KiB
config 188.00 KiB
local 1.20 MiB
College_Scholarship-rs [direct: primary] admin>

```

Figure 7.5

Log Configuration File



```

File Edit View
{"t":{"$date":"2025-04-13T03:03:57.743+01:00"},"s":"I", "c":"CONTROL", "id":20698, "ctx":"thread1","msg":"***** SERVER RESTARTED *****"}
{"t":{"$date":"2025-04-13T03:03:57.745+01:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0
specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-04-13T03:03:58.753+01:00"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}
{"t":{"$date":"2025-04-13T03:03:58.753+01:00"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":"thread1","msg":"Implicit TCP FastOpen unavailable. If TCP FastOpen is required,
set at least one of the related parameters","attr":{"relatedParameters":["tcpFastOpenServer","tcpFastOpenClient","tcpFastOpenQueueSize"]}}
 {"t":{"$date":"2025-04-13T03:03:58.764+01:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"thread1","msg":"Initialized wire
specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":25}, "incomingInternalClient":{"minWireVersion":0,"maxWireVersion":25}, "outgoing":{"minWireVersion":6,"maxWireVersion":25}, "isInternalClient":true}}}
 {"t":{"$date":"2025-04-13T03:03:58.764+01:00"},"s":"I", "c":"CONTROL", "id":7091600, "ctx":"thread1","msg":"Starting TenantMigrationAccessBlockerRegistry"}
 {"t":{"$date":"2025-04-13T03:03:58.765+01:00"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"initandlisten","msg":MongoDB
starting,"attr":{"pid":2508, "port":27017, "dbPath":"C:/Mongo_Project/replicaset/ml/db", "architecture":"64-bit", "host":"Parimal_Sawant"}}
 {"t":{"$date":"2025-04-13T03:03:58.765+01:00"},"s":"I", "c":"CONTROL", "id":23398, "ctx":"initandlisten","msg":Target operating system minimum
version,"attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
 {"t":{"$date":"2025-04-13T03:03:58.765+01:00"},"s":"I", "c":"CONTROL", "id":23403, "ctx":"initandlisten","msg":Build
Info,"attr":{"buildInfo":{"version":"8.0.6","gitVersion":"80f21521ada43df5d613f5d649d7058c6d46277f","modules":[],"allocator":"tcmalloc-gperf","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
 {"t":{"$date":"2025-04-13T03:03:58.765+01:00"},"s":"I", "c":"CONTROL", "id":51765, "ctx":"initandlisten","msg":Operating System,"attr":{"os":{"name":"Microsoft Windows
10","version":"10.0 (build 26100)"}}}

```

Figure 7.6

Each replica set was configured with an individual log file path in its particular configuration file. These log files records activities like startup, authentication, and replication which helps in monitoring the changes and debugging each node separately.

Section 9: Appendices

Screencast 1:

Aggregation Pipeline using MongoDB Shell - <https://youtu.be/32l6Lc4K2rQ>

Screencast 2:

Aggregation Pipeline using MongoDB Atlas Framework - <https://youtu.be/2EZds2fEtOo>

Links to website, youtube video and courses:

1. MongoDB official document on aggregation –
<https://www.mongodb.com/docs/manual/aggregation/>
2. MongoDB official document for CRUD operations –
<https://www.mongodb.com/docs/manual/crud/>
3. Youtube Video for Getting started with MongoDB Atlas –
<https://www.youtube.com/watch?v=bBA9rUdqmgY>
4. LinkedIn learning course MongoDB Essentials Training –
<https://www.linkedin.com/learning>
5. MongoDB University – CRUD Operations
<https://learn.mongodb.com/courses/crud-operations-in-mongodb>
6. MongoDB University – MongoDB Aggregation –
<https://learn.mongodb.com/courses/mongodb-aggregation>

Articles Books or Publications:

1. Chodorow, K. (2013). *MongoDB: The Definitive Guide* (2nd ed.). O'Reilly Media.
2. MongoDB Blog and Articles for future scope - <https://www.mongodb.com/blog/>