

## Day 8 coding assessment

Name: Arpit Mahalle

Output of User Story 1:

### Authors:

Example Query	Purpose	Field(s) Involved
db.authors.find({ name: "J.K. Rowling" })	Lookup by name	name
db.authors.find({ nationality: "British" })	Filter by nationality	nationality
db.authors.find({ birthYear: { \$lt: 1950 } })	Older authors (less frequent)	birthYear

Frequently queried fields from “Authors” collection.: **name, nationality**

### Books:

Example Query	Purpose	Field(s) Involved
db.books.find({ genre: "Fantasy" })	Filter books by genre	genre
db.books.find({ authorId: 1 })	Find all books by an author	authorId
db.books.find({ "ratings.score": { \$gte: 4.5 } })	Get highly-rated books	ratings.score
db.books.find({ publicationYear: 1998 })	Filter by year (less frequent)	publicationYear

Frequently queried fields from “Books” collection: **genre, authorId, ratings.score**

### Users:

Example Query	Purpose	Field(s) Involved
db.users.find({ email: "shyam@example.com" })	Login or lookup by email	email
db.users.find({ joinDate: { \$gte: ISODate("2023-01-01") } })	Filter by join date (less frequent)	joinDate

Frequently queried fields from “Users” collection: **email**.

```
> db.books.createIndex({ genre: 1 });
< genre_1
> db.books.createIndex({ authorId: 1 });
< authorId_1
> db.books.createIndex({ "ratings.score": 1 });
< ratings.score_1
> db.users.createIndex({ email: 1 });
< email_1
```

Index is created on these fields.

### Queries:

```
db.books.dropIndex({ "ratings.score": 1 })
```

```
db.books.getIndexes()
```

```
< [
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { genre: 1 }, name: 'genre_1' },
  { v: 2, key: { authorId: 1 }, name: 'authorId_1' }
]
```

ratings.score Index is dropped and the remaining fields are displayed.

## User story 2 output:

```
> db.books.aggregate([
  { $unwind: "$ratings" }, // Flatten ratings array
  {
    $group: {
      _id: "$title",
      averageRating: { $avg: "$ratings.score" }
    }
  },
  {
    $project: {
      _id: 0,
      title: "$_id",
      averageRating: { $round: ["$averageRating", 2] }
    }
  }
]);
< [
  {
    title: "Harry Potter and the Sorcerer's Stone",
    averageRating: 4.9
  },
  {
    title: 'Harry Potter and the Chamber of Secrets',
    averageRating: 4.8
  },
  {
    title: 'Kafka on the Shore',
    averageRating: 4.5
  },
  {
    title: 'A Game of Thrones',
    averageRating: 4.7
  }
]
```

Calculating the average rating per book using \$unwind, \$group, and \$avg.

```
db.books.aggregate([
  { $unwind: "$ratings" },
  {
    $group: {
      _id: "$title",
      averageRating: { $avg: "$ratings.score" }
    }
  },
  { $sort: { averageRating: -1 } }, // Sort in descending order
  { $limit: 3 }, // Pick top 3
  {
    $project: {
      _id: 0,
      title: "$_id",
      averageRating: 1
    }
  }
]);
{
  averageRating: 4.9,
  title: "Harry Potter and the Sorcerer's Stone"
}
{
  averageRating: 4.8,
  title: 'Harry Potter and the Chamber of Secrets'
}
{
  averageRating: 4.7,
  title: 'A Game of Thrones'
}
```

Retrieving the top 3 highest-rated books.

```
db.books.aggregate([
  {
    $group: {
      _id: "$genre",
      totalBooks: { $sum: 1 }
    }
  },
  {
    $project: {
      _id: 0,
      genre: "$_id",
      totalBooks: 1
    }
  }
]);
{
  totalBooks: 4,
  genre: 'Fantasy'
}
{
  totalBooks: 1,
  genre: 'Magical Realism'
}
```

Counting the number of books published per genre.

```
db.books.aggregate([
  {
    $group: {
      _id: "$authorId",
      booksPublished: { $sum: 1 }
    }
  },
  { $match: { booksPublished: { $gt: 2 } } },
  {
    $lookup: {
      from: "authors",
      localField: "_id",
      foreignField: "_id",
      as: "authorDetails"
    }
  },
  { $unwind: "$authorDetails" },
  {
    $project: {
      _id: 0,
      authorName: "$authorDetails.name",
      booksPublished: 1
    }
  }
]);

```

Find authors who have more than 2 books published: The output is empty because no author has more than 2 books published.

```

db.books.aggregate([
  { $unwind: "$ratings" },
  {
    $group: {
      _id: "$authorId",
      totalRewardPoints: { $sum: "$ratings.score" }
    }
  },
  {
    $lookup: {
      from: "authors",
      localField: "_id",
      foreignField: "_id",
      as: "authorDetails"
    }
  },
  { $unwind: "$authorDetails" },
  {
    $project: {
      _id: 0,
      authorName: "$authorDetails.name",
      totalRewardPoints: { $round: ["$totalRewardPoints", 2] }
    }
  }
]);
{
  authorName: 'Haruki Murakami',
  totalRewardPoints: 4.5
}
{
  authorName: 'J.K. Rowling',
  totalRewardPoints: 9.7
}
{
  authorName: 'George R.R. Martin',
  totalRewardPoints: 4.7
}

```

Displaying the total reward points (sum of all ratings) received by each author.

## User Story 3 output:

The screenshot shows the MongoDB Atlas Project Overview page for 'Project O'. The left sidebar includes sections for Project Overview, Database, Streaming Data, Services, Security, and Application Development. The main area displays 'Cluster0' with options to 'Connect' or 'Edit configuration', and a 'Browse collections' button. A message at the top indicates a new IP address has been added. The right side features a 'Toolbar' with tabs for Performance, Cost, and Resilience.

Free cluster is created. The name of the cluster is “Cluster0”.

The screenshot shows the MongoDB Data Explorer for the 'Authors' collection in 'Cluster0'. The left sidebar lists databases and collections. The main interface shows the schema for the 'Authors' collection, including fields like '\_id', 'name', 'nationality', and 'birthYear'. Three documents are listed in the results table.

_id	name	nationality	birthYear
1	J.K. Rowling	British	1965
2	George R.R. Martin	American	1949
3	Haruki Murakami	Japanese	1949

Data in “Authors” collection is inserted.

The screenshot shows the MongoDB Cloud Data Explorer interface. On the left, a sidebar menu includes Cluster, Overview, Data Explorer (which is selected and highlighted in green), Real Time, Cluster Metrics, Query Insights, Performance Advisor, Online Archive, Command Line Tools, Infrastructure as Code, and SHORTCUTS (Search & Vector Search). The main panel displays the 'BookVerseCloudDB.Books' collection. It shows three document snippets:

```
_id: 1001
title: "Harry Potter and the Sorcerer's Stone"
genre: "Fantasy"
publicationYear: 1997
authorId: 1
+ ratings: Array (2)

_id: 1002
title: "Harry Potter and the Chamber of Secrets"
genre: "Fantasy"
publicationYear: 1998
authorId: 1
+ ratings: Array (1)

_id: 1003
title: "A Game of Thrones"
genre: "Fantasy"
publicationYear: 1996
authorId: 2
+ ratings: Array (1)
```

Below the documents, there is a note: "System Status: All Good". At the bottom, the status bar shows "©2025 MongoDB, Inc. Status Terms Privacy Alios Blog Contact Sales".

Data in “Books” collection is inserted.

The screenshot shows the MongoDB Cloud Data Explorer interface. The sidebar menu is identical to the previous screenshot. The main panel displays the 'BookVerseCloudDB.Users' collection. It shows four document snippets:

```
_id: 101
name: "Shyam Kumar"
email: "shyam@example.com"
joinDate: "2023-01-01T00:00:00.000+00:00"

_id: 102
name: "Priya Shah"
email: "priya@example.com"
joinDate: "2023-05-20T00:00:00.000+00:00"

_id: 103
name: "Rahul Verma"
email: "rahul@example.com"
joinDate: "2023-05-20T00:00:00.000+00:00"
```

Below the documents, there is a note: "System Status: All Good". At the bottom, the status bar shows "©2025 MongoDB, Inc. Status Terms Privacy Alios Blog Contact Sales".

Data in “Users” collection is inserted.

The screenshot shows the MongoDB Compass interface. At the top, there's a header bar with the title 'MongoDB Compass - cluster0|jqmnevq.mongodb.net/Databases'. Below the header is a navigation bar with 'Connections', 'Edit', 'View', and 'Help' options. The main area is divided into two sections: 'Compass' on the left and a database list on the right.

**Compass** sidebar:

- My Queries
- Data Modeling
- CONNECTIONS [4] (with a search bar)
- cluster0|jqmnevq.mongodb.net (selected)
- BookVerseCloudDB
- BookVerseDB
- EventDB
- GreenPulse
- Item
- admin
- config (selected)
- demoDB
- local

**Databases** list:

Database name	Storage size	Collections	Indexes
BookVerseCloudDB	61.44 kB	3	3
BookVerseDB	188.42 kB	6	9
EventDB	73.73 kB	2	2
GreenPulse	24.58 kB	1	1
Item	73.73 kB	2	4
admin	36.84 kB	1	1
config	0 B	0	0
demoDB	-	0	-
local	38.06 kB	1	1
	-	0	-

Buttons at the top right include 'Open MongoDB shell', 'Create database', and 'Refresh'.

Connected my cluster using the connection string in MongoDB Compass.