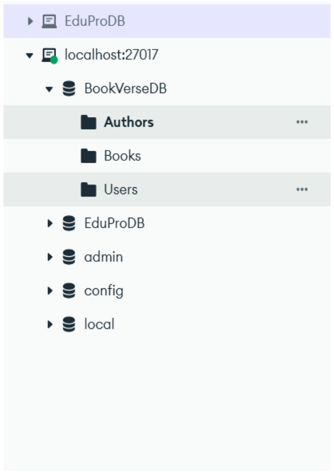
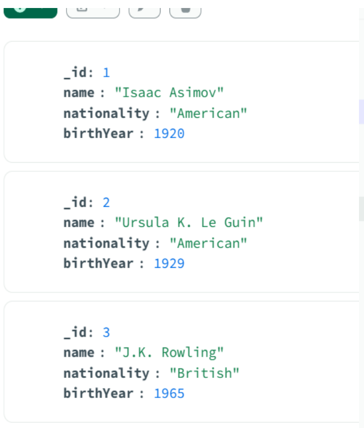
USER STORY 1:

After creating the Database name with BookVerseDB:

I make tables Authors, Books and User and inserted data in it.

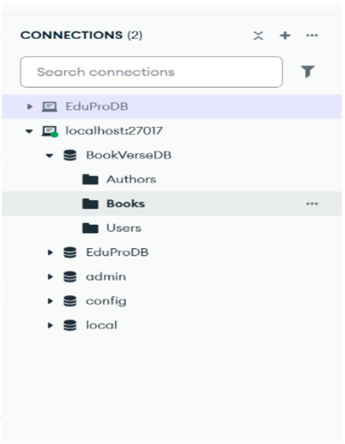


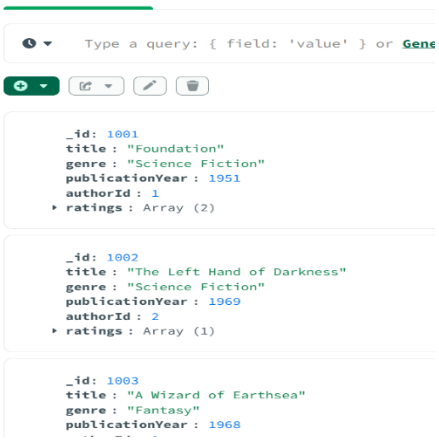


```
_id: 1
name : "Isaac Asimov"
nationality : "American"
birthYear : 1920

_id: 2
name : "Ursula K. Le Guin"
nationality : "American"
birthYear : 1929

_id: 3
name : "J.K. Rowling"
nationality : "British"
birthYear : 1965
```





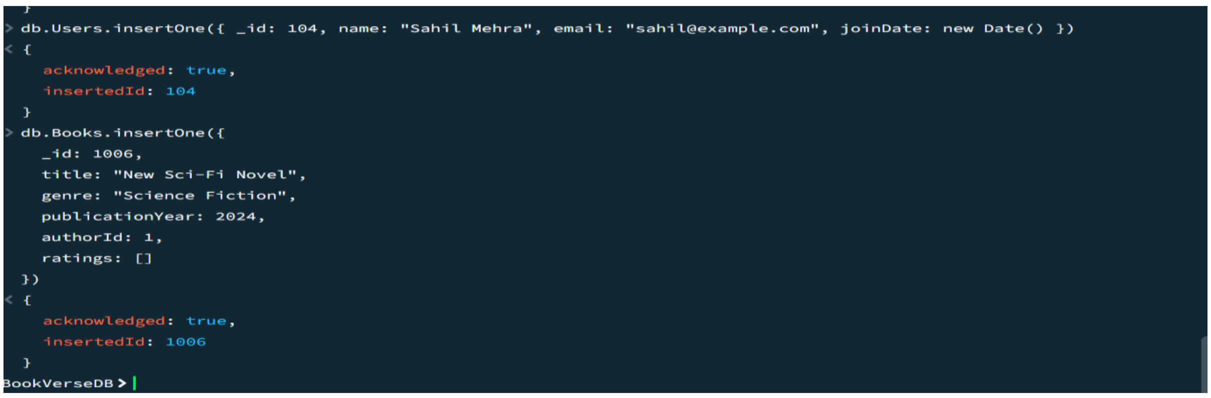
```
_id: 1001
title : "Foundation"
genre : "Science Fiction"
publicationYear : 1951
authorId : 1
ratings : Array (2)

_id: 1002
title : "The Left Hand of Darkness"
genre : "Science Fiction"
publicationYear : 1969
authorId : 2
ratings : Array (1)

_id: 1003
title : "A Wizard of Earthsea"
genre : "Fantasy"
publicationYear : 1968
```

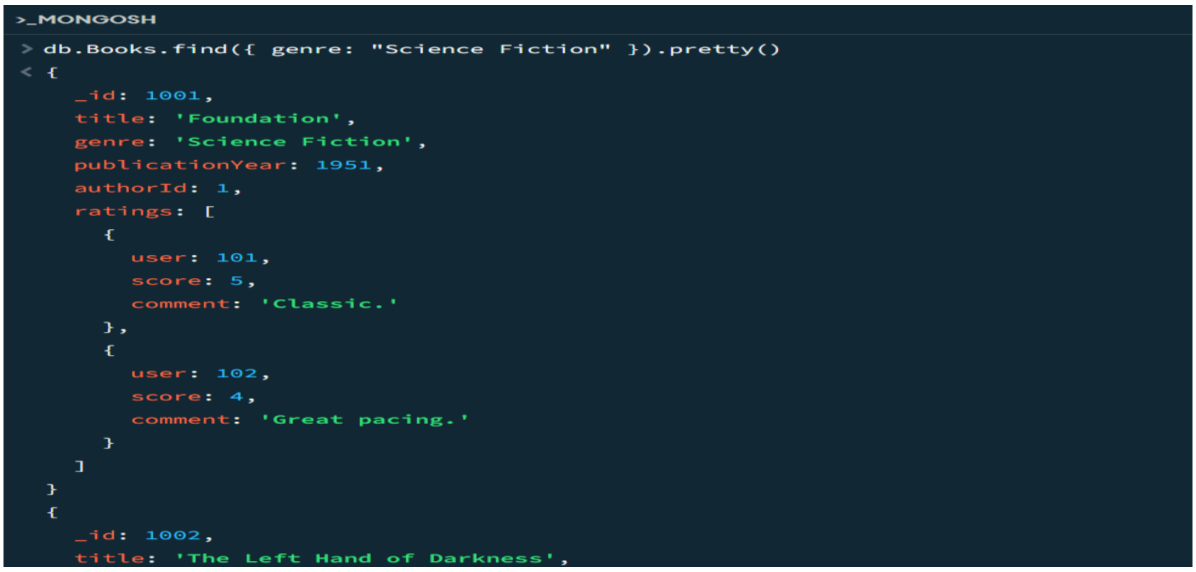
USER STORY 2:

1. Create — Insert new user and new book



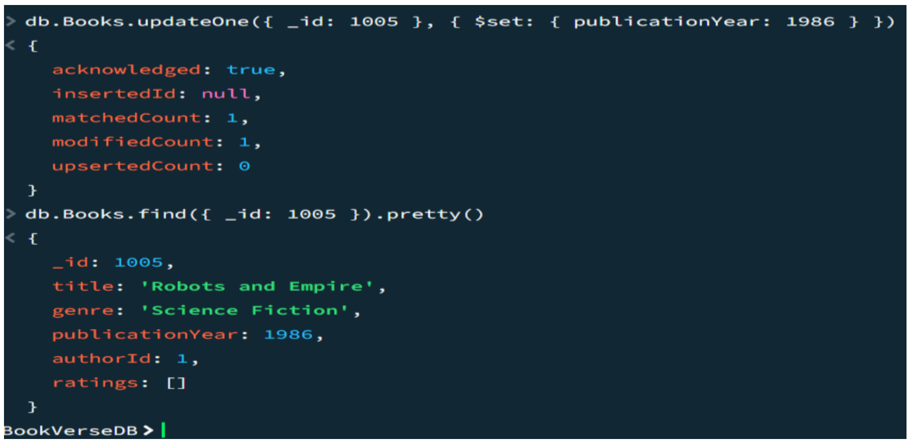
```
> db.Users.insertOne({ _id: 104, name: "Sahil Mehra", email: "sahil@example.com", joinDate: new Date() })
< {
    acknowledged: true,
    insertedId: 104
  }
> db.Books.insertOne({
    _id: 1006,
    title: "New Sci-Fi Novel",
    genre: "Science Fiction",
    publicationYear: 2024,
    authorId: 1,
    ratings: []
  })
< {
    acknowledged: true,
    insertedId: 1006
  }
BookVerseDB >
```

2. Read — Retrieve all books of genre "Science Fiction"



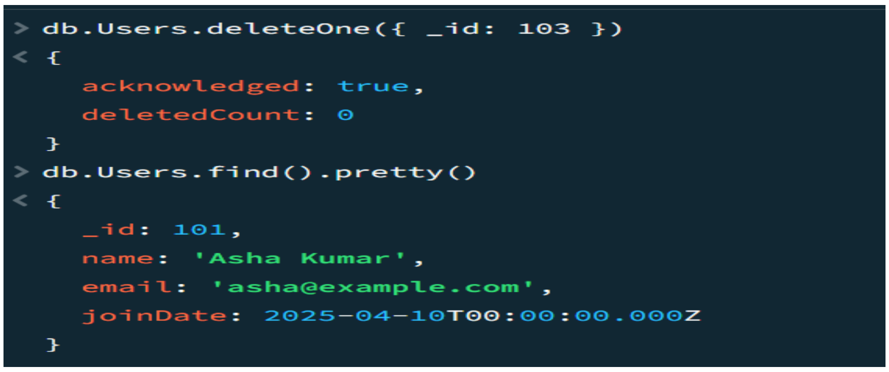
```
>_MONGOSH
> db.Books.find({ genre: "Science Fiction" }).pretty()
< {
    _id: 1001,
    title: 'Foundation',
    genre: 'Science Fiction',
    publicationYear: 1951,
    authorId: 1,
    ratings: [
      {
        user: 101,
        score: 5,
        comment: 'Classic.'
      },
      {
        user: 102,
        score: 4,
        comment: 'Great pacing.'
      }
    ]
  }
  {
    _id: 1002,
    title: 'The Left Hand of Darkness',
```

3. Update — Update the publication Year of one book



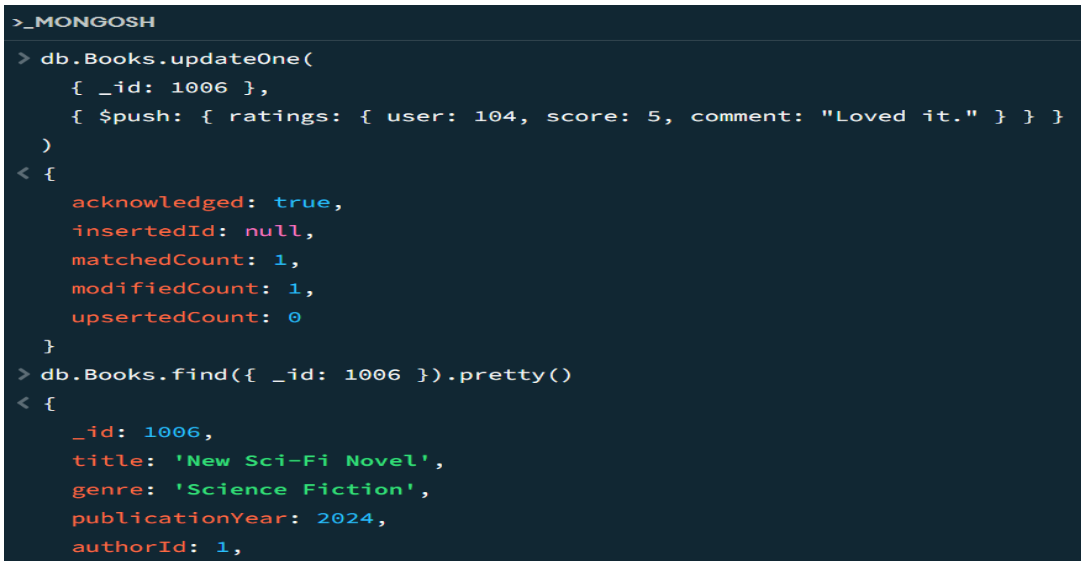
```
> db.Books.updateOne({ _id: 1005 }, { $set: { publicationYear: 1986 } })
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.Books.find({ _id: 1005 }).pretty()
< {
    _id: 1005,
    title: 'Robots and Empire',
    genre: 'Science Fiction',
    publicationYear: 1986,
    authorId: 1,
    ratings: []
  }
BookVerseDB >
```

4. Delete — Delete one user record



```
> db.Users.deleteOne({ _id: 103 })
< {
    acknowledged: true,
    deletedCount: 0
  }
> db.Users.find().pretty()
< {
    _id: 101,
    name: 'Asha Kumar',
    email: 'asha@example.com',
    joinDate: 2025-04-10T00:00:00.000Z
  }
```

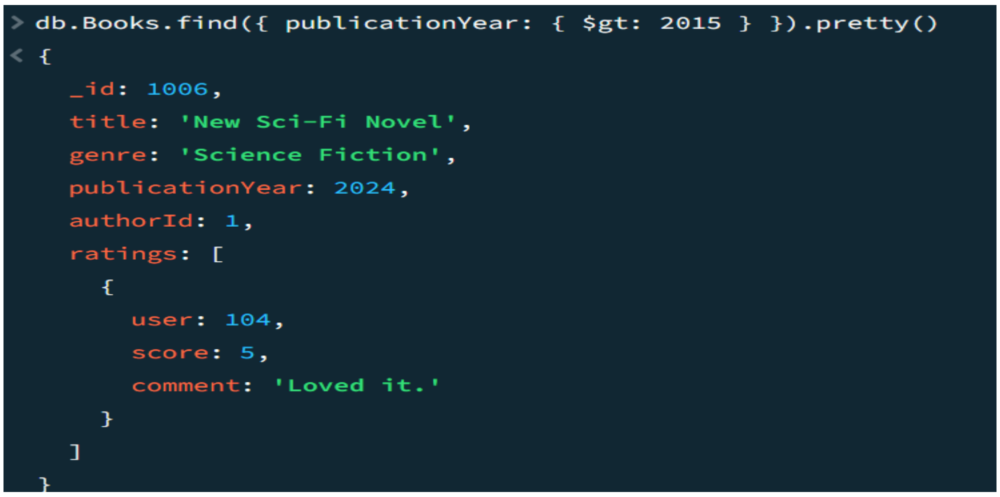
5. Update — Add a new rating to a book using $push



```
>_MONGOSH
> db.Books.updateOne(
    { _id: 1006 },
    { $push: { ratings: { user: 104, score: 5, comment: "Loved it." } } }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.Books.find({ _id: 1006 }).pretty()
< {
    _id: 1006,
    title: 'New Sci-Fi Novel',
    genre: 'Science Fiction',
    publicationYear: 2024,
    authorId: 1,
```

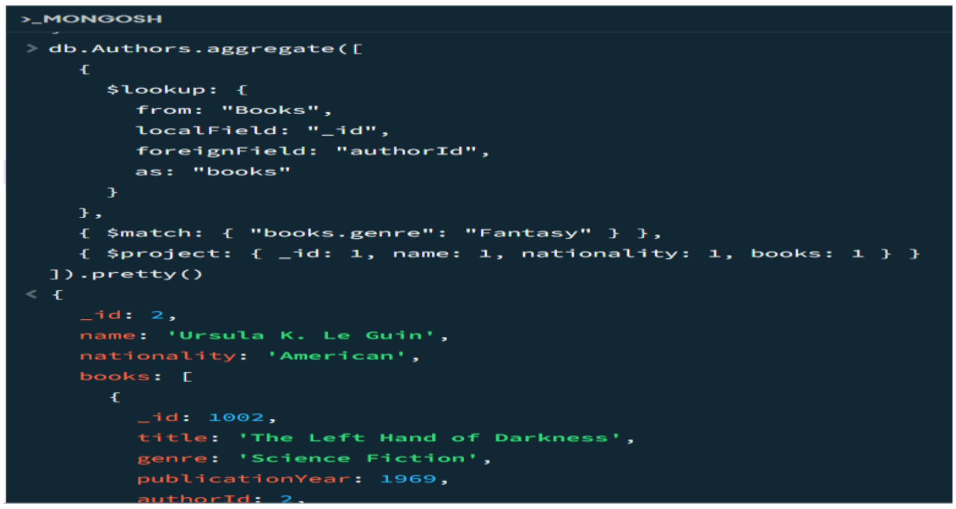
USER STORY 3

1. Retrieve all books published after 2015



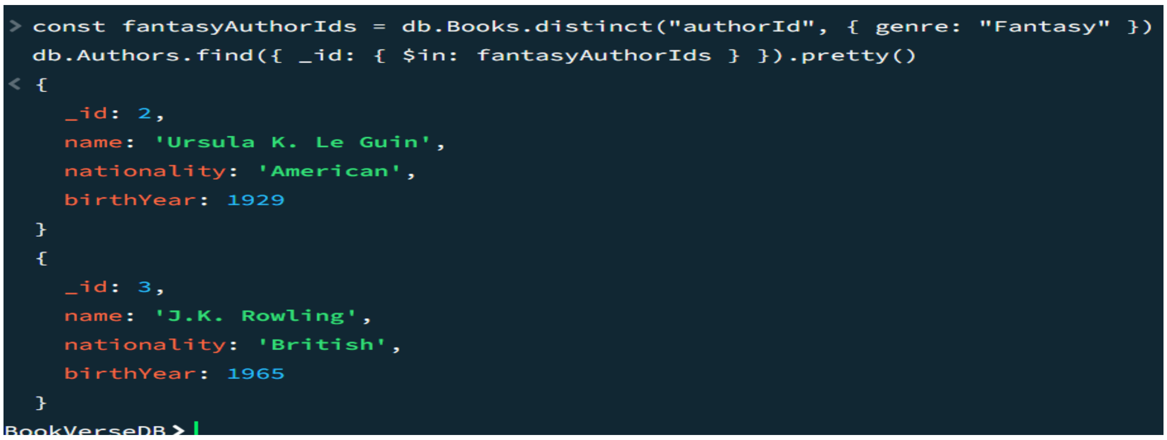
```
> db.Books.find({ publicationYear: { $gt: 2015 } }).pretty()
< {
    _id: 1006,
    title: 'New Sci-Fi Novel',
    genre: 'Science Fiction',
    publicationYear: 2024,
    authorId: 1,
    ratings: [
      {
        user: 104,
        score: 5,
        comment: 'Loved it.'
      }
    ]
  }
```

2. Find authors who have written books in the "Fantasy" genre
   A. Using Aggregation



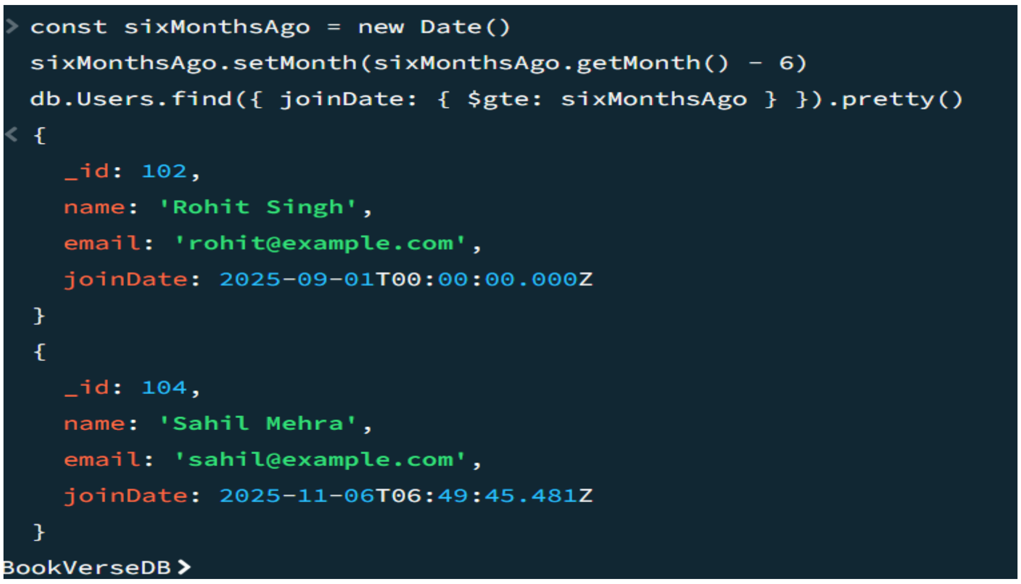
```
>_MONGOSH
> db.Authors.aggregate([
    {
      $lookup: {
        from: "Books",
        localField: "_id",
        foreignField: "authorId",
        as: "books"
      }
    },
    { $match: { "books.genre": "Fantasy" } },
    { $project: { _id: 1, name: 1, nationality: 1, books: 1 } }
  ]).pretty()
< {
    _id: 2,
    name: 'Ursula K. Le Guin',
    nationality: 'American',
    books: [
      {
        _id: 1002,
        title: 'The Left Hand of Darkness',
        genre: 'Science Fiction',
        publicationYear: 1969,
        authorId: 2,
```

B . find distinct authorIds from Books and then lookup:



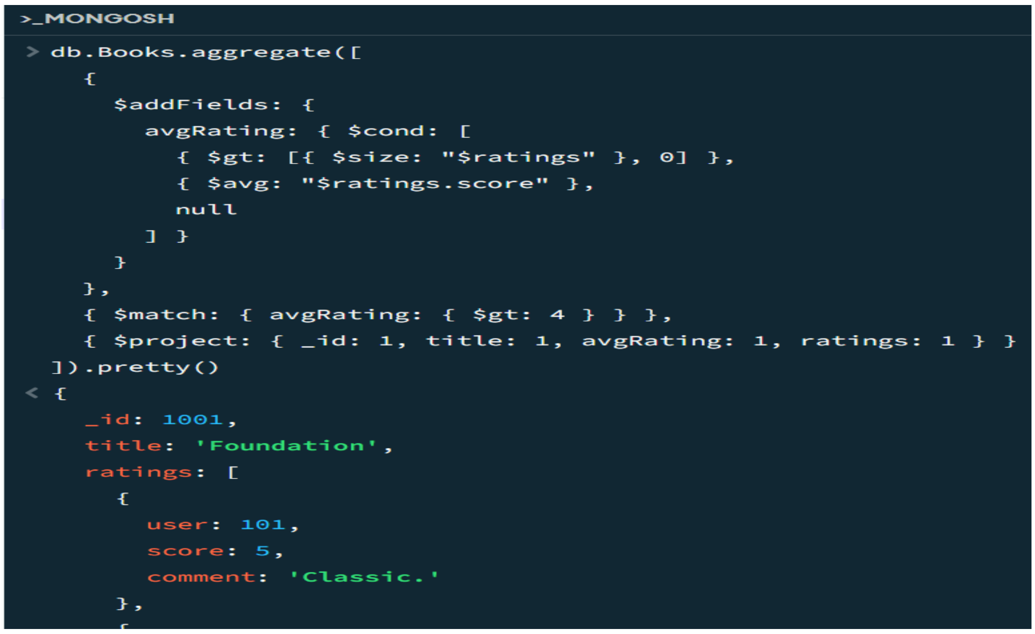
```
> const fantasyAuthorIds = db.Books.distinct("authorId", { genre: "Fantasy" })
  db.Authors.find({ _id: { $in: fantasyAuthorIds } }).pretty()
< {
    _id: 2,
    name: 'Ursula K. Le Guin',
    nationality: 'American',
    birthYear: 1929
  }
  {
    _id: 3,
    name: 'J.K. Rowling',
    nationality: 'British',
    birthYear: 1965
  }
BookVerseDB >
```

3.Retrieve all users who joined within the last 6 months



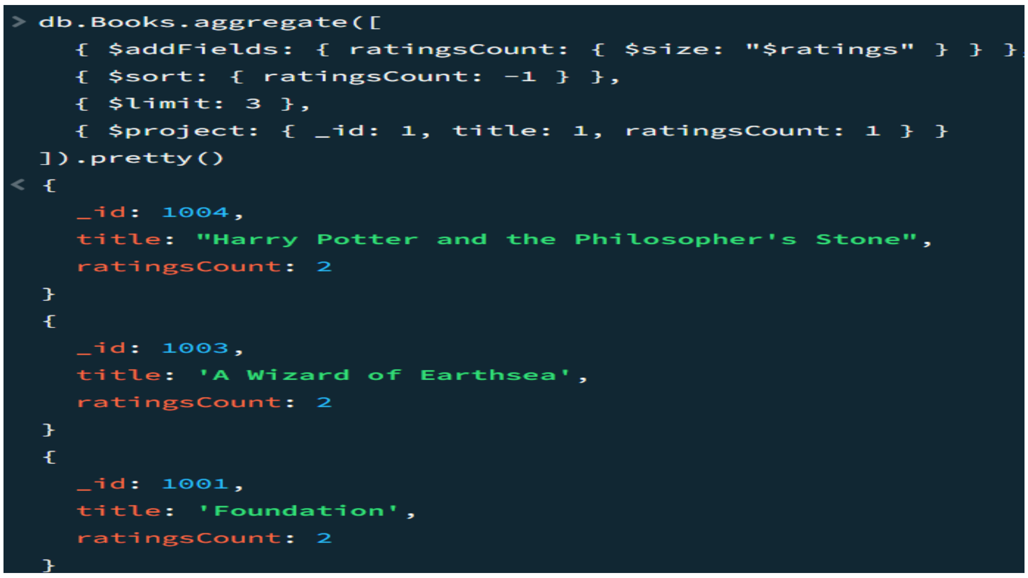
```
> const sixMonthsAgo = new Date()
  sixMonthsAgo.setMonth(sixMonthsAgo.getMonth() - 6)
  db.Users.find({ joinDate: { $gte: sixMonthsAgo } }).pretty()
< {
    _id: 102,
    name: 'Rohit Singh',
    email: 'rohit@example.com',
    joinDate: 2025-09-01T00:00:00.000Z
  }
  {
    _id: 104,
    name: 'Sahil Mehra',
    email: 'sahil@example.com',
    joinDate: 2025-11-06T06:49:45.481Z
  }
BookVerseDB >
```

4.Find books with an average rating greater than 4



```
>_MONGOSH
> db.Books.aggregate([
    {
      $addFields: {
        avgRating: { $cond: [
          { $gt: [{ $size: "$ratings" }, 0] },
          { $avg: "$ratings.score" },
          null
        ] }
      }
    },
    { $match: { avgRating: { $gt: 4 } } },
    { $project: { _id: 1, title: 1, avgRating: 1, ratings: 1 } }
  ]).pretty()
< {
    _id: 1001,
    title: 'Foundation',
    ratings: [
      {
        user: 101,
        score: 5,
        comment: 'Classic.'
      },
```

Bonus 1 — Top 3 most-rated books



```
> db.Books.aggregate([
    { $addFields: { ratingsCount: { $size: "$ratings" } } },
    { $sort: { ratingsCount: -1 } },
    { $limit: 3 },
    { $project: { _id: 1, title: 1, ratingsCount: 1 } }
  ]).pretty()
< {
    _id: 1004,
    title: "Harry Potter and the Philosopher's Stone",
    ratingsCount: 2
  }
  {
    _id: 1003,
    title: 'A Wizard of Earthsea',
    ratingsCount: 2
  }
  {
    _id: 1001,
    title: 'Foundation',
    ratingsCount: 2
  }
```