Day 7 Coding Assignment — MongoDB Data Modeling and CRUD Operations

// all Syntaxes to be executed in mongosh Shell

UserStory 1

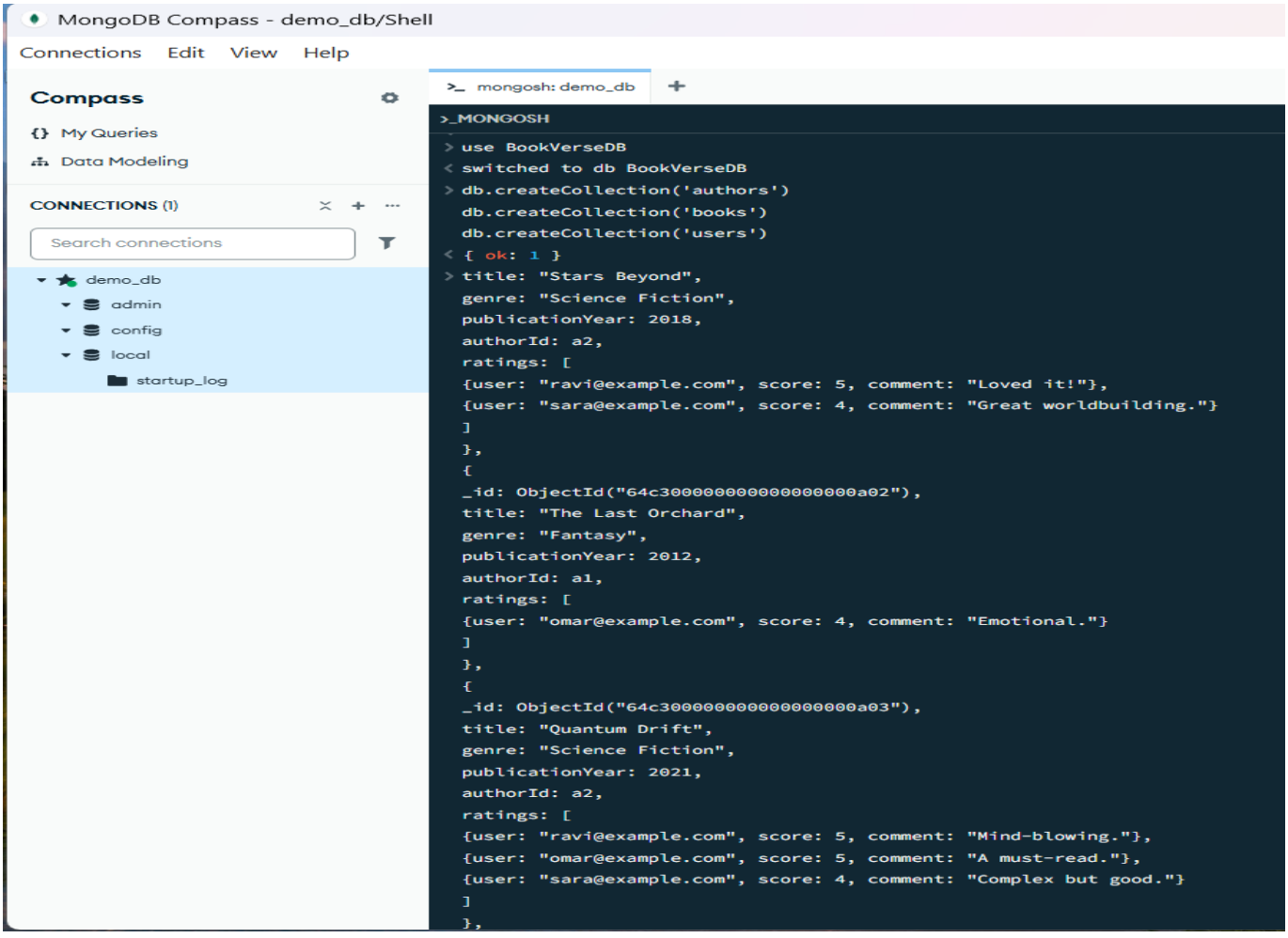
Code

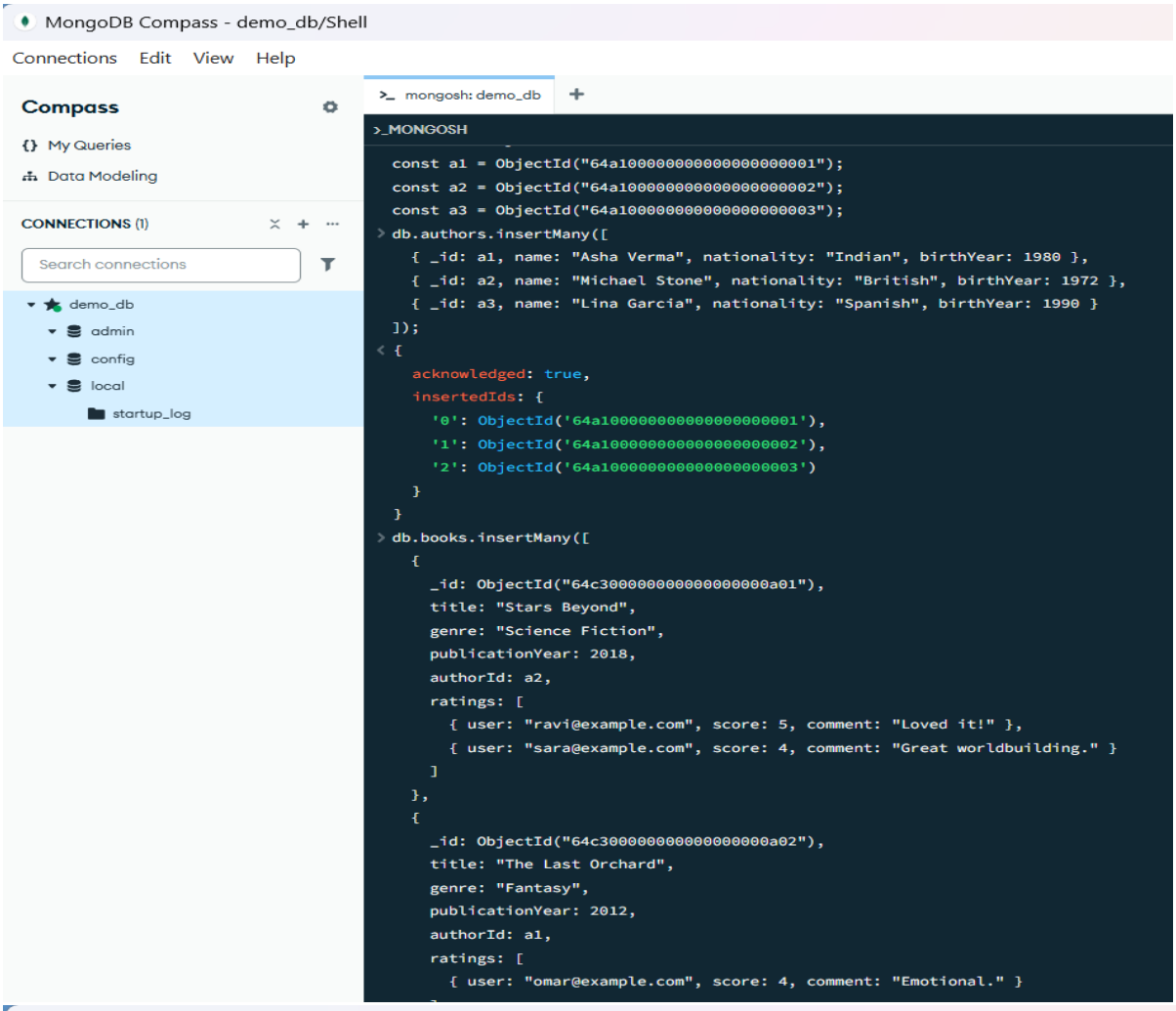
```
use BookVerseDB
db.createCollection('authors')
db.createCollection('books')
db.createCollection('users')
const a1 = ObjectId("64a100000000000000000001");
const a2 = ObjectId("64a100000000000000000002");
const a3 = ObjectId("64a100000000000000000003");
db.authors.insertMany([
 { _id: a1, name: "Asha Verma", nationality: "Indian", birthYear: 1980 },
 { _id: a2, name: "Michael Stone", nationality: "British", birthYear: 1972 },
 { _id: a3, name: "Lina Garcia", nationality: "Spanish", birthYear: 1990 }
]);
db.books.insertMany([
 {
  _id: ObjectId("64c300000000000000000a01"),
  title: "Stars Beyond",
  genre: "Science Fiction",
  publicationYear: 2018,
  authorId: a2,
  ratings: [
   { user: "ravi@example.com", score: 5, comment: "Loved it!" },
   { user: "sara@example.com", score: 4, comment: "Great worldbuilding." }
  ]
 },
 {
  _id: ObjectId("64c300000000000000000a02"),
```

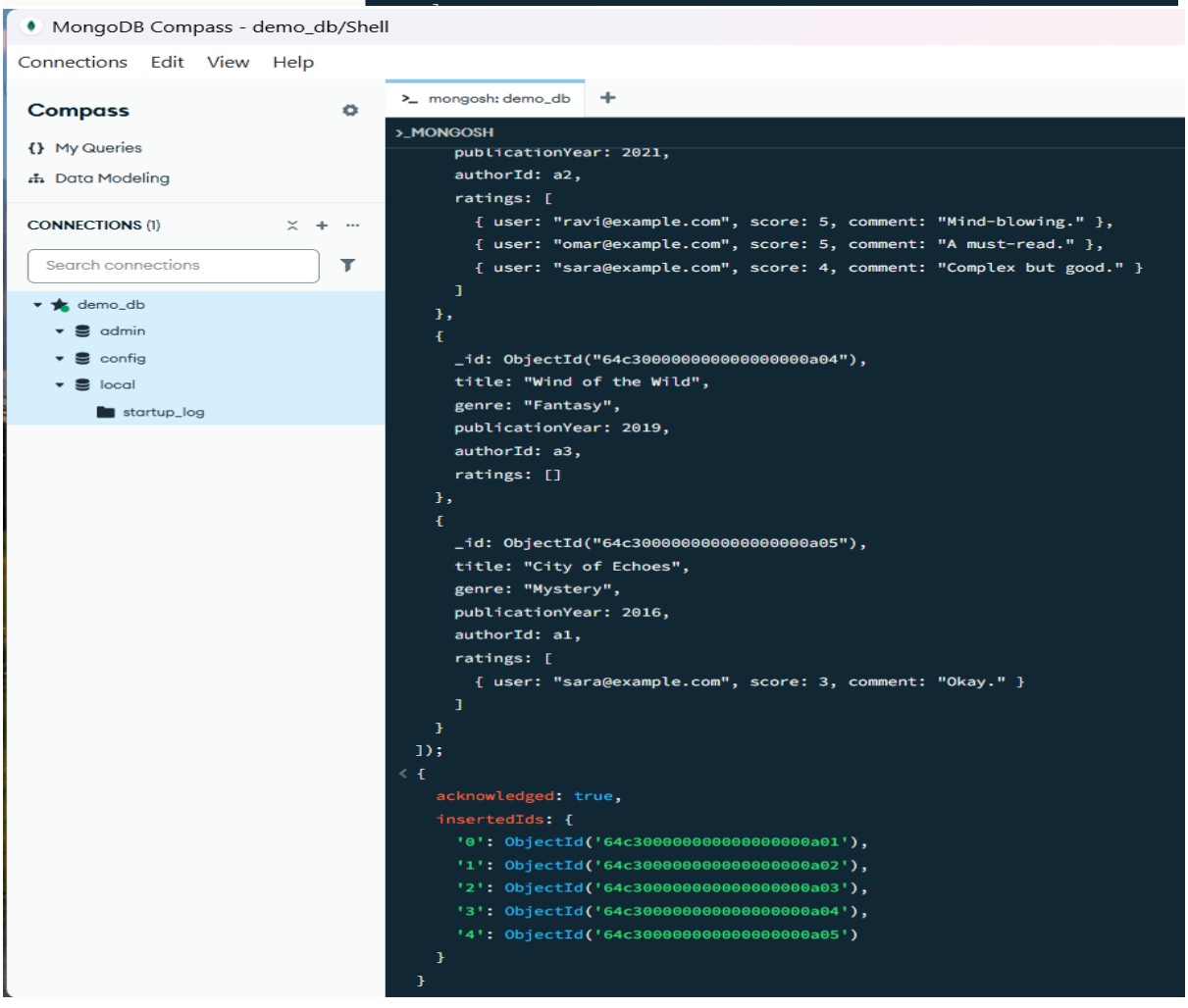
```
  title: "The Last Orchard",
  genre: "Fantasy",
  publicationYear: 2012,
  authorId: a1,
  ratings: [
    { user: "omar@example.com", score: 4, comment: "Emotional." }
  ]
},
{
  _id: ObjectId("64c300000000000000a03"),
  title: "Quantum Drift",
  genre: "Science Fiction",
  publicationYear: 2021,
  authorId: a2,
  ratings: [
    { user: "ravi@example.com", score: 5, comment: "Mind-blowing." },
    { user: "omar@example.com", score: 5, comment: "A must-read." },
    { user: "sara@example.com", score: 4, comment: "Complex but good." }
  ]
},
{
  _id: ObjectId("64c300000000000000a04"),
  title: "Wind of the Wild",
  genre: "Fantasy",
  publicationYear: 2019,
  authorId: a3,
  ratings: []
},
{
  _id: ObjectId("64c300000000000000a05"),
  title: "City of Echoes",
  genre: "Mystery",
```

```
    publicationYear: 2016,
    authorId: a1,
    ratings: [
      { user: "sara@example.com", score: 3, comment: "Okay." }
    ]
  }
]);
db.authors.find().pretty();
db.books.find().pretty();
```

Output:-




MongoDB Compass - demo_db/Shell
Connections Edit View Help
Compass
My Queries
Data Modeling
CONNECTIONS (1)
Search connections
demo_db
admin
config
local
startup_log
mongosh: demo_db
>_MONGOSH
const a1 = ObjectId("64a1000000000000000000001");
const a2 = ObjectId("64a1000000000000000000002");
const a3 = ObjectId("64a1000000000000000000003");
db.authors.insertMany([
{ _id: a1, name: "Asha Verma", nationality: "Indian", birthYear: 1980 },
{ _id: a2, name: "Michael Stone", nationality: "British", birthYear: 1972 },
{ _id: a3, name: "Lina Garcia", nationality: "Spanish", birthYear: 1990 }
]);
{
acknowledged: true,
insertedIds: {
'0': ObjectId('64a1000000000000000000001'),
'1': ObjectId('64a1000000000000000000002'),
'2': ObjectId('64a1000000000000000000003')
}
}
db.books.insertMany([
{
_id: ObjectId("64c3000000000000000a01"),
title: "Stars Beyond",
genre: "Science Fiction",
publicationYear: 2018,
authorId: a2,
ratings: [
{ user: "ravi@example.com", score: 5, comment: "Loved it!" },
{ user: "sara@example.com", score: 4, comment: "Great worldbuilding." }
]
},
{
_id: ObjectId("64c3000000000000000a02"),
title: "The Last Orchard",
genre: "Fantasy",
publicationYear: 2012,
authorId: a1,
ratings: [
{ user: "omar@example.com", score: 4, comment: "Emotional." }


MongoDB Compass - demo_db/Shell
Connections Edit View Help
Compass
My Queries
Data Modeling
CONNECTIONS (1)
Search connections
demo_db
admin
config
local
startup_log
mongosh: demo_db
>_MONGOSH
publicationYear: 2021,
authorId: a2,
ratings: [
{ user: "ravi@example.com", score: 5, comment: "Mind-blowing." },
{ user: "omar@example.com", score: 5, comment: "A must-read." },
{ user: "sara@example.com", score: 4, comment: "Complex but good." }
]
},
{
_id: ObjectId("64c3000000000000000a04"),
title: "Wind of the Wild",
genre: "Fantasy",
publicationYear: 2019,
authorId: a3,
ratings: []
},
{
_id: ObjectId("64c3000000000000000a05"),
title: "City of Echoes",
genre: "Mystery",
publicationYear: 2016,
authorId: a1,
ratings: [
{ user: "sara@example.com", score: 3, comment: "Okay." }
]
}
]);
{
acknowledged: true,
insertedIds: {
'0': ObjectId('64c3000000000000000a01'),
'1': ObjectId('64c3000000000000000a02'),
'2': ObjectId('64c3000000000000000a03'),
'3': ObjectId('64c3000000000000000a04'),
'4': ObjectId('64c3000000000000000a05')
}
}

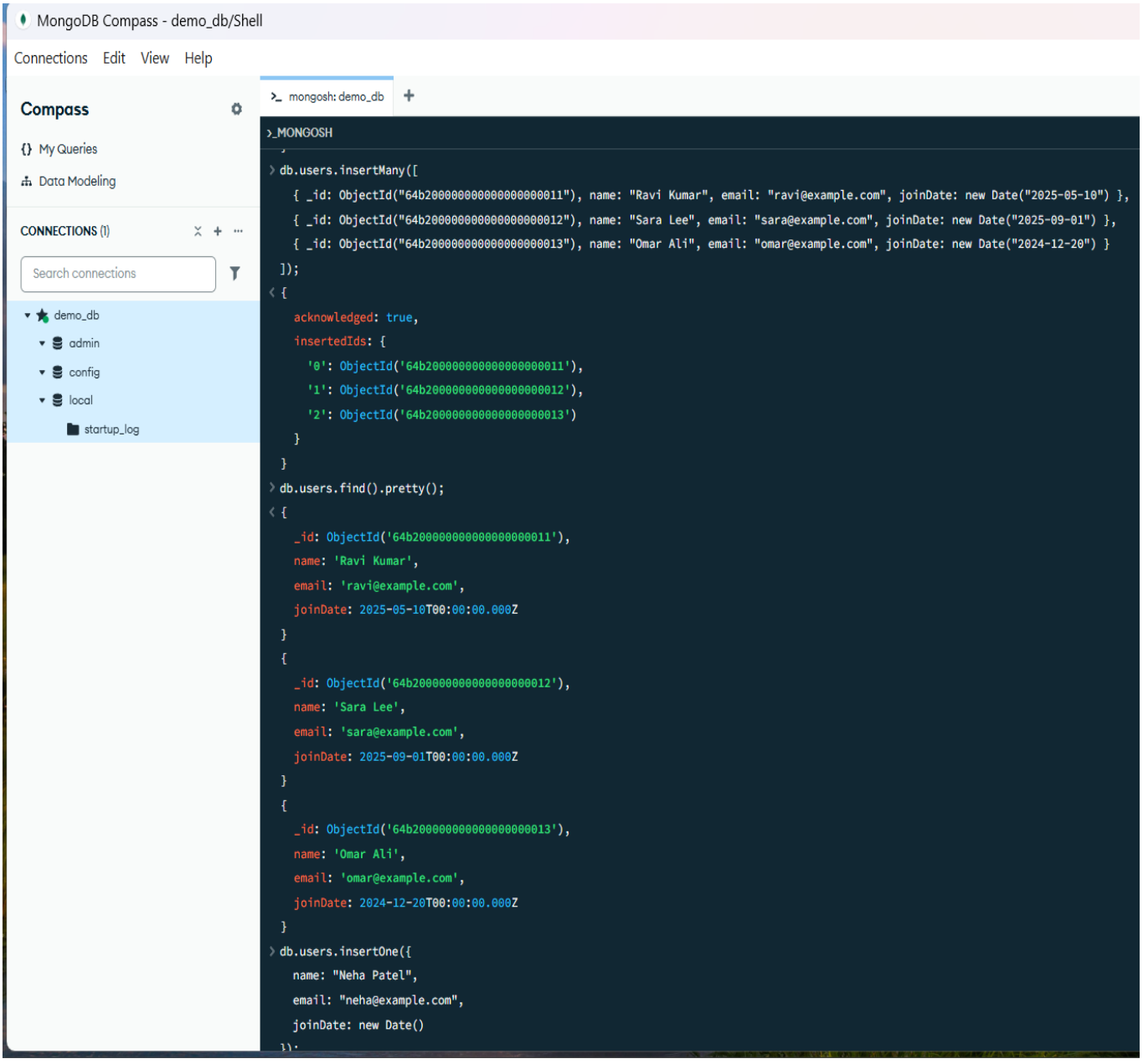
UserStory 2

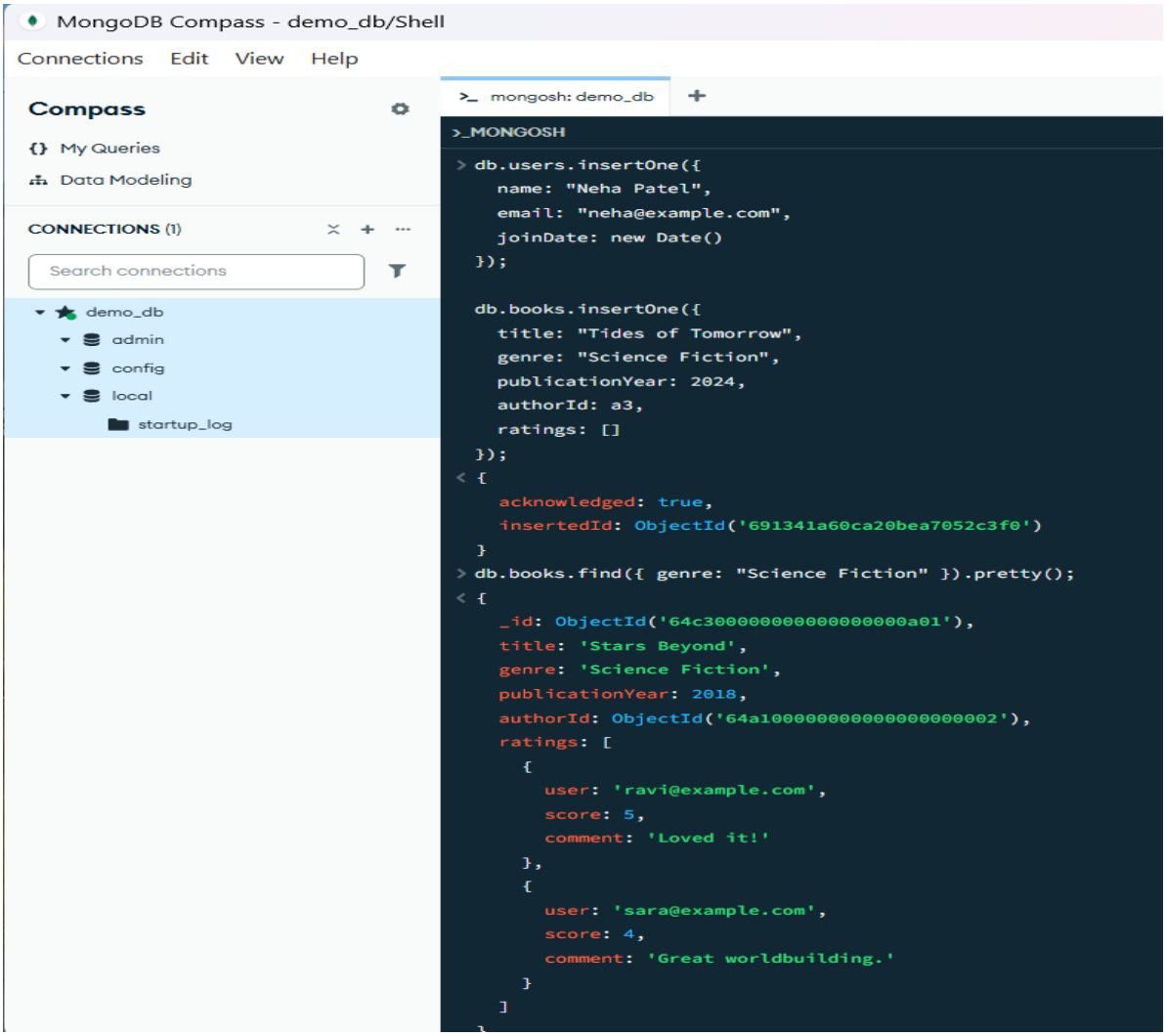
Code:-

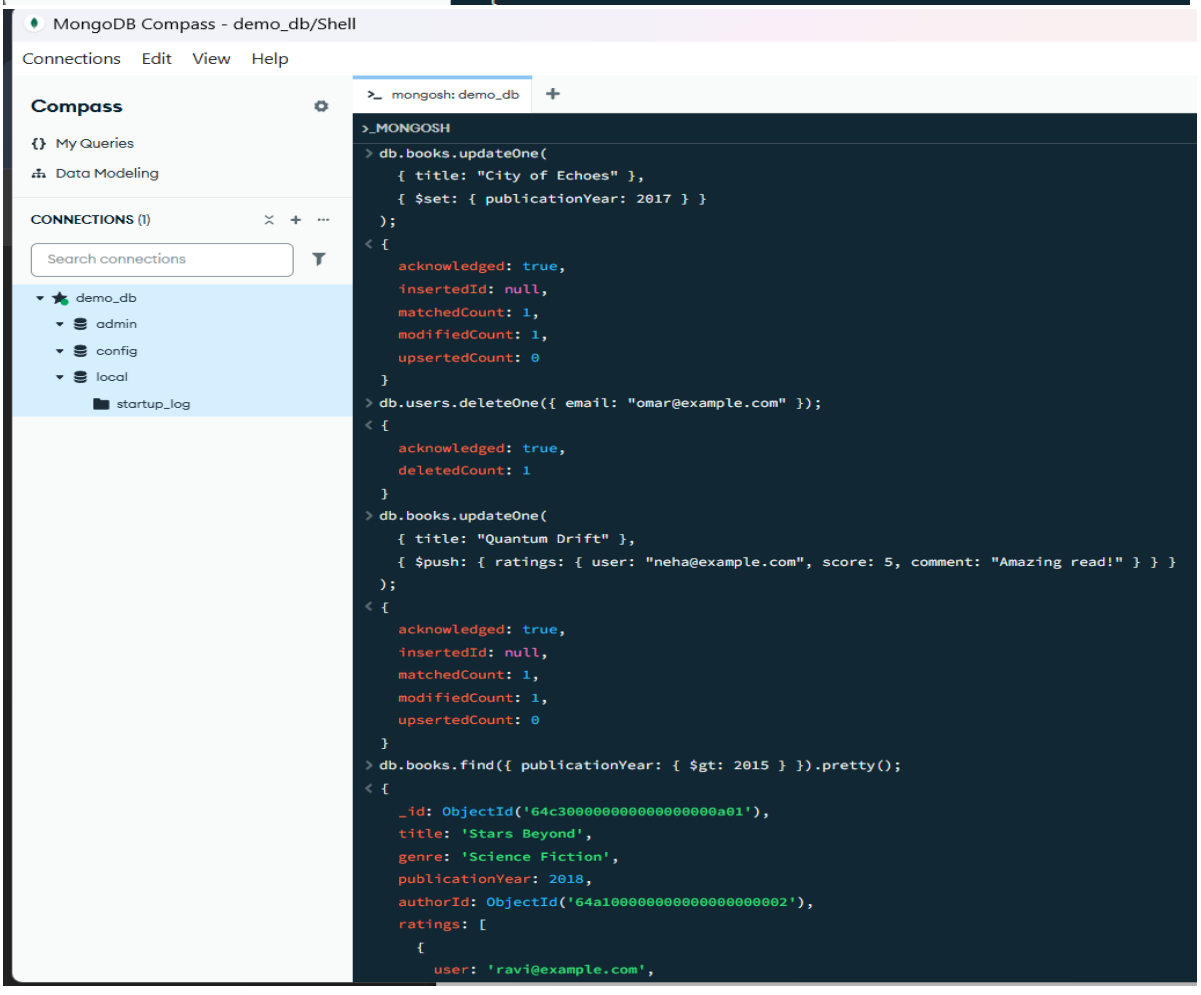
```
db.users.insertMany([

  { _id: ObjectId("64b20000000000000000011"), name: "Ravi Kumar", email: "ravi@example.com", joinDate: new Date("2025-05-10") },

  { _id: ObjectId("64b20000000000000000012"), name: "Sara Lee", email: "sara@example.com", joinDate: new Date("2025-09-01") },

  { _id: ObjectId("64b20000000000000000013"), name: "Omar Ali", email: "omar@example.com", joinDate: new Date("2024-12-20") }

]);

db.users.find().pretty();

db.users.insertOne({

  name: "Neha Patel",

  email: "neha@example.com",

  joinDate: new Date()

});


db.books.insertOne({

  title: "Tides of Tomorrow",

  genre: "Science Fiction",

  publicationYear: 2024,

  authorId: a3,

  ratings: []

});

db.books.find({ genre: "Science Fiction" }).pretty();

db.books.updateOne(
```

```
  { title: "City of Echoes" },
  { $set: { publicationYear: 2017 } }
);
db.users.deleteOne({ email: "omar@example.com" });
db.books.updateOne(
  { title: "Quantum Drift" },
  { $push: { ratings: { user: "neha@example.com", score: 5, comment: "Amazing read!" } } }
);
```

Output:-




MongoDB Compass - demo_db/Shell
Connections Edit View Help
Compass
My Queries
Data Modeling
CONNECTIONS (1)
Search connections
demo_db
admin
config
local
startup_log
mongosh: demo_db
>_MONGOSH
> db.users.insertOne({
name: "Neha Patel",
email: "neha@example.com",
joinDate: new Date()
});
db.books.insertOne({
title: "Tides of Tomorrow",
genre: "Science Fiction",
publicationYear: 2024,
authorId: a3,
ratings: []
});
< {
acknowledged: true,
insertedId: ObjectId('691341a60ca20bea7052c3f0')
}
> db.books.find({ genre: "Science Fiction" }).pretty();
< {
_id: ObjectId('64c300000000000000000a01'),
title: 'Stars Beyond',
genre: 'Science Fiction',
publicationYear: 2018,
authorId: ObjectId('64a100000000000000000002'),
ratings: [
{
user: 'ravi@example.com',
score: 5,
comment: 'Loved it!'
},
{
user: 'sara@example.com',
score: 4,
comment: 'Great worldbuilding.'
}
]

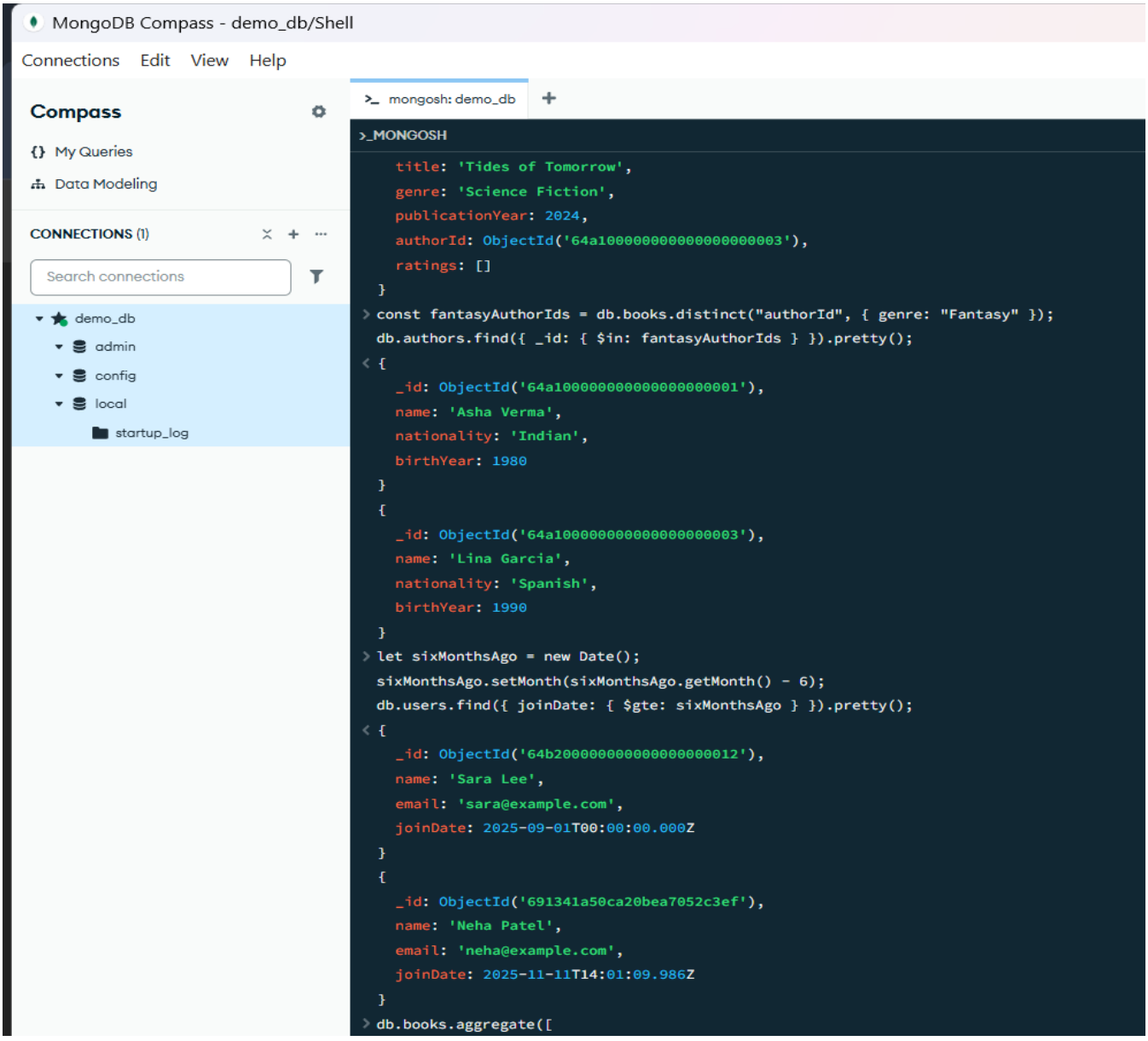

MongoDB Compass - demo_db/Shell
Connections Edit View Help
Compass
My Queries
Data Modeling
CONNECTIONS (1)
Search connections
demo_db
admin
config
local
startup_log
mongosh: demo_db
>_MONGOSH
> db.books.updateOne(
{ title: "City of Echoes" },
{ $set: { publicationYear: 2017 } }
);
< {
acknowledged: true,
insertedId: null,
matchedCount: 1,
modifiedCount: 1,
upsertedCount: 0
}
> db.users.deleteOne({ email: "omar@example.com" });
< {
acknowledged: true,
deletedCount: 1
}
> db.books.updateOne(
{ title: "Quantum Drift" },
{ $push: { ratings: { user: "neha@example.com", score: 5, comment: "Amazing read!" } } }
);
< {
acknowledged: true,
insertedId: null,
matchedCount: 1,
modifiedCount: 1,
upsertedCount: 0
}
> db.books.find({ publicationYear: { $gt: 2015 } }).pretty();
< {
_id: ObjectId('64c300000000000000000a01'),
title: 'Stars Beyond',
genre: 'Science Fiction',
publicationYear: 2018,
authorId: ObjectId('64a100000000000000000002'),
ratings: [
{
user: 'ravi@example.com',

UserStory 3

Code:-

```
db.books.find({ publicationYear: { $gt: 2015 } }).pretty();

const fantasyAuthorIds = db.books.distinct("authorId", { genre: "Fantasy" });

db.authors.find({ _id: { $in: fantasyAuthorIds } }).pretty();

let sixMonthsAgo = new Date();

sixMonthsAgo.setMonth(sixMonthsAgo.getMonth() - 6);

db.users.find({ joinDate: { $gte: sixMonthsAgo } }).pretty();
```

Output:-

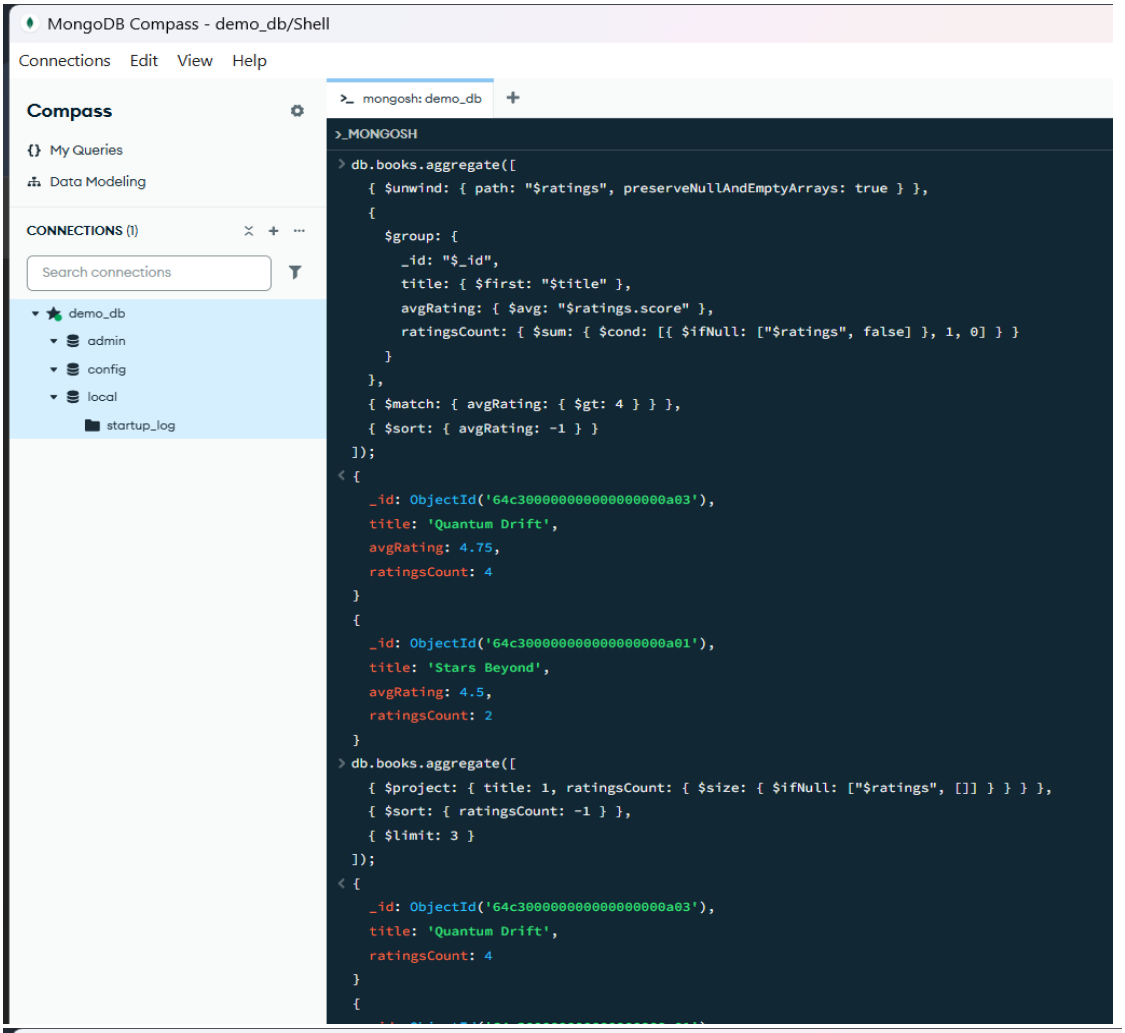


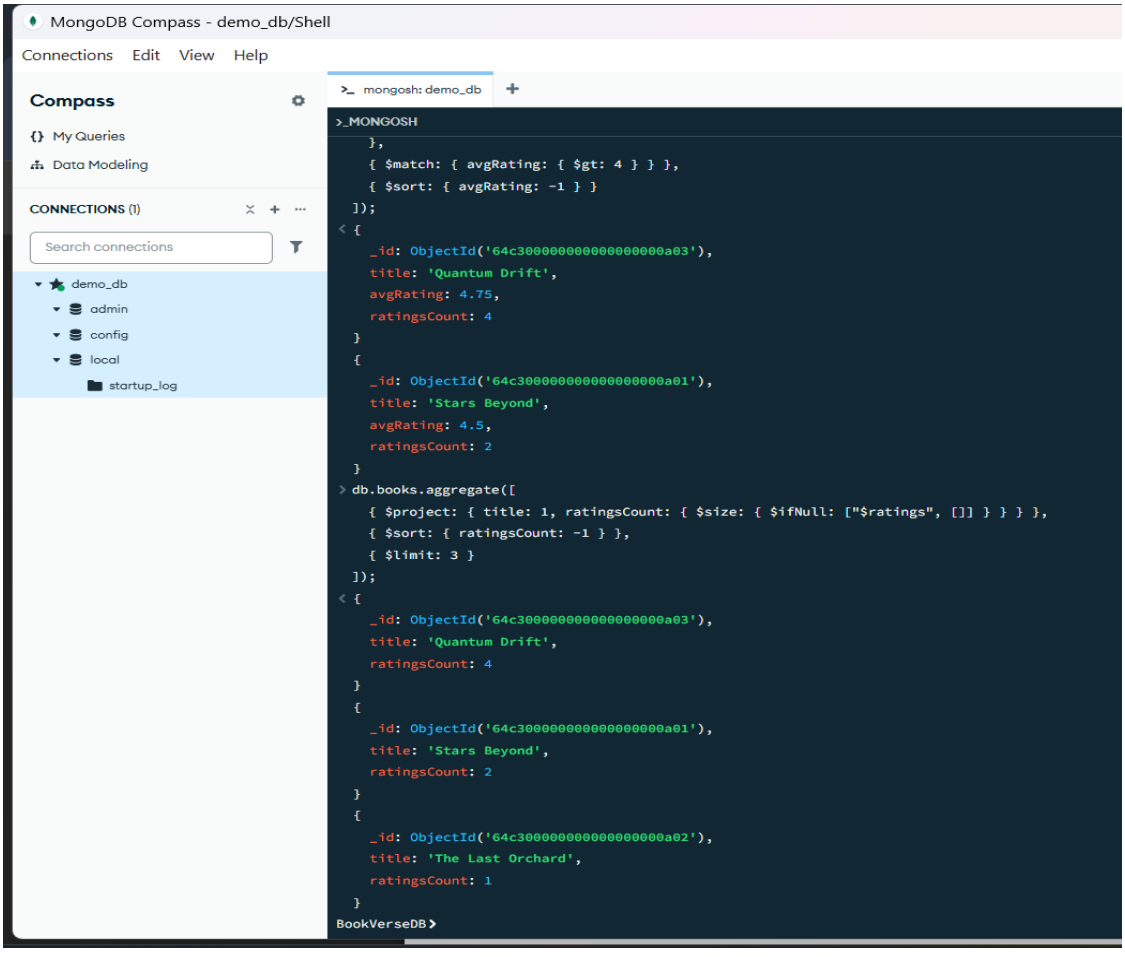
Bonus Challenge (Optional)

Code :-

```
db.books.aggregate([
 { $unwind: { path: "$ratings", preserveNullAndEmptyArrays: true } },
 {
  $group: {
   _id: "$_id",
   title: { $first: "$title" },
   avgRating: { $avg: "$ratings.score" },
   ratingsCount: { $sum: { $cond: [{ $ifNull: ["$ratings", false] }, 1, 0] } }
  }
 },
 { $match: { avgRating: { $gt: 4 } } },
 { $sort: { avgRating: -1 } }
]);
db.books.aggregate([
 { $project: { title: 1, ratingsCount: { $size: { $ifNull: ["$ratings", []] } } } },
 { $sort: { ratingsCount: -1 } },
 { $limit: 3 }
]);
```

Output:-


MongoDB Compass - demo_db/Shell
Connections Edit View Help
Compass
My Queries
Data Modeling
CONNECTIONS (1)
Search connections
demo_db
admin
config
local
startup_log
mongosh: demo_db
>_MONGOSH
> db.books.aggregate([
{ $unwind: { path: "$ratings", preserveNullAndEmptyArrays: true } },
{
$group: {
_id: "$_id",
title: { $first: "$title" },
avgRating: { $avg: "$ratings.score" },
ratingsCount: { $sum: { $cond: [{ $ifNull: ["$ratings", false] }, 1, 0] } }
}
},
{ $match: { avgRating: { $gt: 4 } } },
{ $sort: { avgRating: -1 } }
]);
< {
_id: ObjectId('64c30000000000000000a03'),
title: 'Quantum Drift',
avgRating: 4.75,
ratingsCount: 4
}
{
_id: ObjectId('64c30000000000000000a01'),
title: 'Stars Beyond',
avgRating: 4.5,
ratingsCount: 2
}
> db.books.aggregate([
{ $project: { title: 1, ratingsCount: { $size: { $ifNull: ["$ratings", []] } } } },
{ $sort: { ratingsCount: -1 } },
{ $limit: 3 }
]);
< {
_id: ObjectId('64c30000000000000000a03'),
title: 'Quantum Drift',
ratingsCount: 4
}
{


MongoDB Compass - demo_db/Shell
Connections Edit View Help
Compass
My Queries
Data Modeling
CONNECTIONS (1)
Search connections
demo_db
admin
config
local
startup_log
mongosh: demo_db
>_MONGOSH
},
{ $match: { avgRating: { $gt: 4 } } },
{ $sort: { avgRating: -1 } }
]);
< {
_id: ObjectId('64c30000000000000000a03'),
title: 'Quantum Drift',
avgRating: 4.75,
ratingsCount: 4
}
{
_id: ObjectId('64c30000000000000000a01'),
title: 'Stars Beyond',
avgRating: 4.5,
ratingsCount: 2
}
> db.books.aggregate([
{ $project: { title: 1, ratingsCount: { $size: { $ifNull: ["$ratings", []] } } } },
{ $sort: { ratingsCount: -1 } },
{ $limit: 3 }
]);
< {
_id: ObjectId('64c30000000000000000a03'),
title: 'Quantum Drift',
ratingsCount: 4
}
{
_id: ObjectId('64c30000000000000000a01'),
title: 'Stars Beyond',
ratingsCount: 2
}
{
_id: ObjectId('64c30000000000000000a02'),
title: 'The Last Orchard',
ratingsCount: 1
}
BookVerseDB>

script using Node.js and Mongoose to insert data into the collections

Folder Setup

1. Inside the folder bookverse-seed.
2. Inside that folder, there are two files:
    - package.json
    - seed.js

Run the Seeder

In the Terminal

>node seed.js

You will see

> Data successfully seeded into BookVerseDB

Execute this is Mongosh Shell

>use BookVerseDB

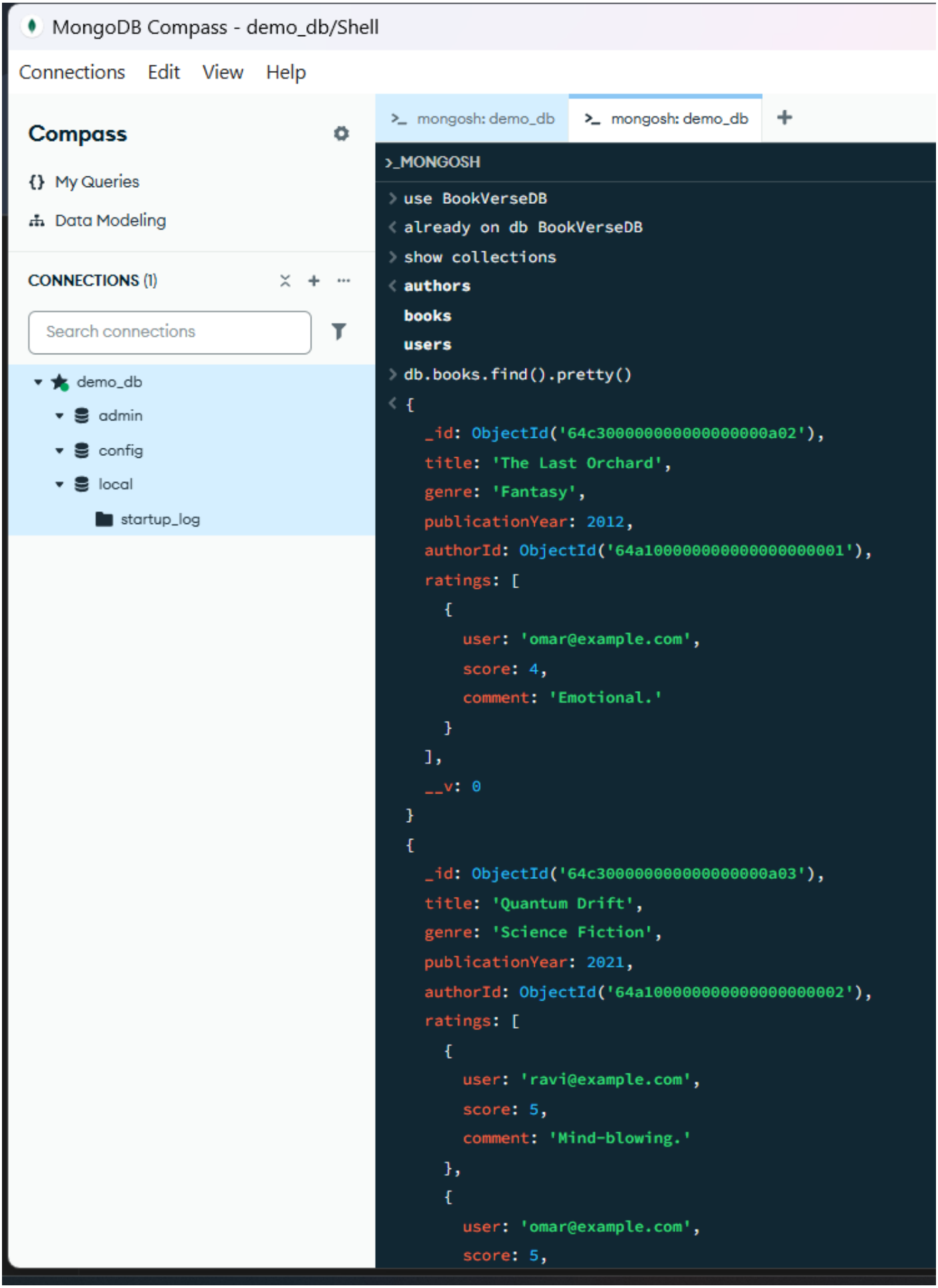
>show collections

>db.books.find().pretty();

Output:-

This script seeds all three collections:

- **authors** (3 documents)
- **books** (5 documents)
- **users** (3 documents)


MongoDB Compass - demo_db/Shell
Connections Edit View Help
Compass
My Queries
Data Modeling
CONNECTIONS (1)
Search connections
demo_db
admin
config
local
startup_log
mongosh: demo_db
mongosh: demo_db
>_MONGOSH
> use BookVerseDB
< already on db BookVerseDB
> show collections
< authors
books
users
> db.books.find().pretty()
< {
_id: ObjectId('64c300000000000000000a02'),
title: 'The Last Orchard',
genre: 'Fantasy',
publicationYear: 2012,
authorId: ObjectId('64a100000000000000000001'),
ratings: [
{
user: 'omar@example.com',
score: 4,
comment: 'Emotional.'
}
],
__v: 0
}
{
_id: ObjectId('64c300000000000000000a03'),
title: 'Quantum Drift',
genre: 'Science Fiction',
publicationYear: 2021,
authorId: ObjectId('64a100000000000000000002'),
ratings: [
{
user: 'ravi@example.com',
score: 5,
comment: 'Mind-blowing.'
},
{
user: 'omar@example.com',
score: 5,