

# Day 8 Coding Assignment — Indexing, Aggregation, and MongoDB Atlas

## User Story 1 — Indexing and Query Optimization

### Code:-

```
// create an index  
db.books.createIndex({ genre: 1 });  
  
db.books.createIndex({ authorId: 1 });  
  
db.books.createIndex({ "ratings.score": 1 });  
  
//before and after index  
db.books.find({ genre: "Fantasy" }).explain("executionStats");  
db.books.find({ genre: "Fantasy" }).explain("executionStats");  
  
// drop an index  
db.books.getIndexes();  
db.books.dropIndex("genre_1");
```

### **output:-**

MongoDB Compass - demo\_db/Shell

Connections Edit View Help

Compass

My Queries Data Modeling

CONNECTIONS (1)

demo\_db BookVerseDB admin config local startup\_log

Search connections

```
>_ mongosh: demo_db +  
>_ MONGOSH  
> use BookVerseDB  
< switched to db BookVerseDB  
> db.books.createIndex({ genre: 1 });  
< genre_1  
> db.books.createIndex({ authorId: 1 });  
  
< authorId_1  
> db.books.createIndex({ "ratings.score": 1 });  
< ratings.score_1  
> db.books.find({ genre: "Fantasy" }).explain("executionStats");  
< {  
  explainVersion: '1',  
  queryPlanner: {  
    namespace: 'BookVerseDB.books',  
    parsedQuery: {  
      genre: {  
        '$eq': 'Fantasy'  
      }  
    },  
    indexFilterSet: false,  
    queryHash: 'B2E493C0',  
    planCacheShapeHash: 'B2E493C0',  
    planCacheKey: '30842E12',  
    optimizationTimeMillis: 20,  
    maxIndexedOrSolutionsReached: false,  
    maxIndexedAndSolutionsReached: false,  
    maxScansToExplodeReached: false,  
    prunedSimilarIndexes: false,  
    winningPlan: {  
      isCached: false,  
      stage: 'FETCH',  
      inputStage: {  
        stage: 'IXSCAN',  
        keyPattern: {  
          genre: 1  
        },  
        indexName: 'genre_1',  
        isMultiKey: false,  
        multiKeyPaths: {  
          genre: []  
        },  
        isUnique: false,  
        isSparse: false,  
        isPartial: false,  
        indexVersion: 2,  
        direction: 'forward',  
        indexBounds: {  
          genre: [  
            ["Fantasy", "Fantasy"]  
          ]  
        }  
      }  
    }  
  }  
}
```

MongoDB Compass - demo\_db/Shell

Connections Edit View Help

Compass

My Queries Data Modeling

CONNECTIONS (1)

demo\_db BookVerseDB admin config local startup\_log

Search connections

```
>_ mongosh: demo_db +  
>_ MONGOSH  
> db.books.find({ genre: "Fantasy" }).explain("executionStats");  
< {  
  explainVersion: '1',  
  queryPlanner: {  
    namespace: 'BookVerseDB.books',  
    parsedQuery: {  
      genre: {  
        '$eq': 'Fantasy'  
      }  
    },  
    indexFilterSet: false,  
    queryHash: 'B2E493C0',  
    planCacheShapeHash: 'B2E493C0',  
    planCacheKey: '30842E12',  
    optimizationTimeMillis: 0,  
    maxIndexedOrSolutionsReached: false,  
    maxIndexedAndSolutionsReached: false,  
    maxScansToExplodeReached: false,  
    prunedSimilarIndexes: false,  
    winningPlan: {  
      isCached: false,  
      stage: 'FETCH',  
      inputStage: {  
        stage: 'IXSCAN',  
        keyPattern: {  
          genre: 1  
        },  
        indexName: 'genre_1',  
        isMultiKey: false,  
        multiKeyPaths: {  
          genre: []  
        },  
        isUnique: false,  
        isSparse: false,  
        isPartial: false,  
        indexVersion: 2,  
        direction: 'forward',  
        indexBounds: {  
          genre: [  
            ["Fantasy", "Fantasy"]  
          ]  
        }  
      }  
    }  
  }  
}
```

The screenshot shows the MongoDB Compass application interface. On the left, the sidebar displays 'Connections' with one connection named 'demo\_db' selected. The main area is a terminal window titled 'mongosh: demo\_db' showing the following command history:

```

>_MONGOSH
{
  "queryShapeHash": "D4DED17F18C07BF496812771203621E52E645DA95474C657EEDB5E7837142B79",
  "command": {
    "find": "books",
    "filter": {
      "genre": "Fantasy"
    },
    "$db": "BookVerseDB"
  },
  "serverInfo": {
    "host": "KIRAN-PAVAN-PAMIDI-4HU1QCJ2",
    "port": 27017,
    "version": "8.2.1",
    "gitVersion": "3312bdcf28aa65f5f930005e21c2cb130f648b8c3"
  },
  "serverParameters": {
    "internalQueryFacetBufferSizeBytes": 104857600,
    "internalQueryFacetMaxOutputDocSizeBytes": 104857600,
    "internalLookupStageIntermediateDocumentMaxSizeBytes": 104857600,
    "internalDocumentSourceGroupMaxMemoryBytes": 104857600,
    "internalQueryMaxBlockingSortMemoryUsageBytes": 104857600,
    "internalQueryProhibitBlockingMergeOnMongoS": 0,
    "internalQueryMaxAddToSetBytes": 104857600,
    "internalDocumentSourceSetWindowFieldsMaxMemoryBytes": 104857600,
    "internalQueryFrameworkControl": "trySbeRestricted",
    "internalQueryPlannerIgnoreIndexWithCollationForRegex": 1
  },
  "ok": 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' },
  { v: 2, key: { 'ratings.score': 1 }, name: 'ratings.score_1' }
]
> db.books.dropIndex("genre_1");
< { nIndexesWas: 4, ok: 1 }

BookVerseDB>

```

## User Story 2 — Aggregation Framework

**Code:-**

```

db.books.aggregate([
  { $unwind: "$ratings" },
  {
    $group: {
      _id: "$_id",
      title: { $first: "$title" },
      avgRating: { $avg: "$ratings.score" }
    }
  }
])

```

```
}

]);


db.books.aggregate([
  { $unwind: "$ratings" },
  {
    $group: {
      _id: "$_id",
      title: { $first: "$title" },
      avgRating: { $avg: "$ratings.score" }
    }
  },
  { $sort: { avgRating: -1 } },
  { $limit: 3 }
]);


db.books.aggregate([
  {
    $group: {
      _id: "$genre",
      bookCount: { $sum: 1 }
    }
  }
]);

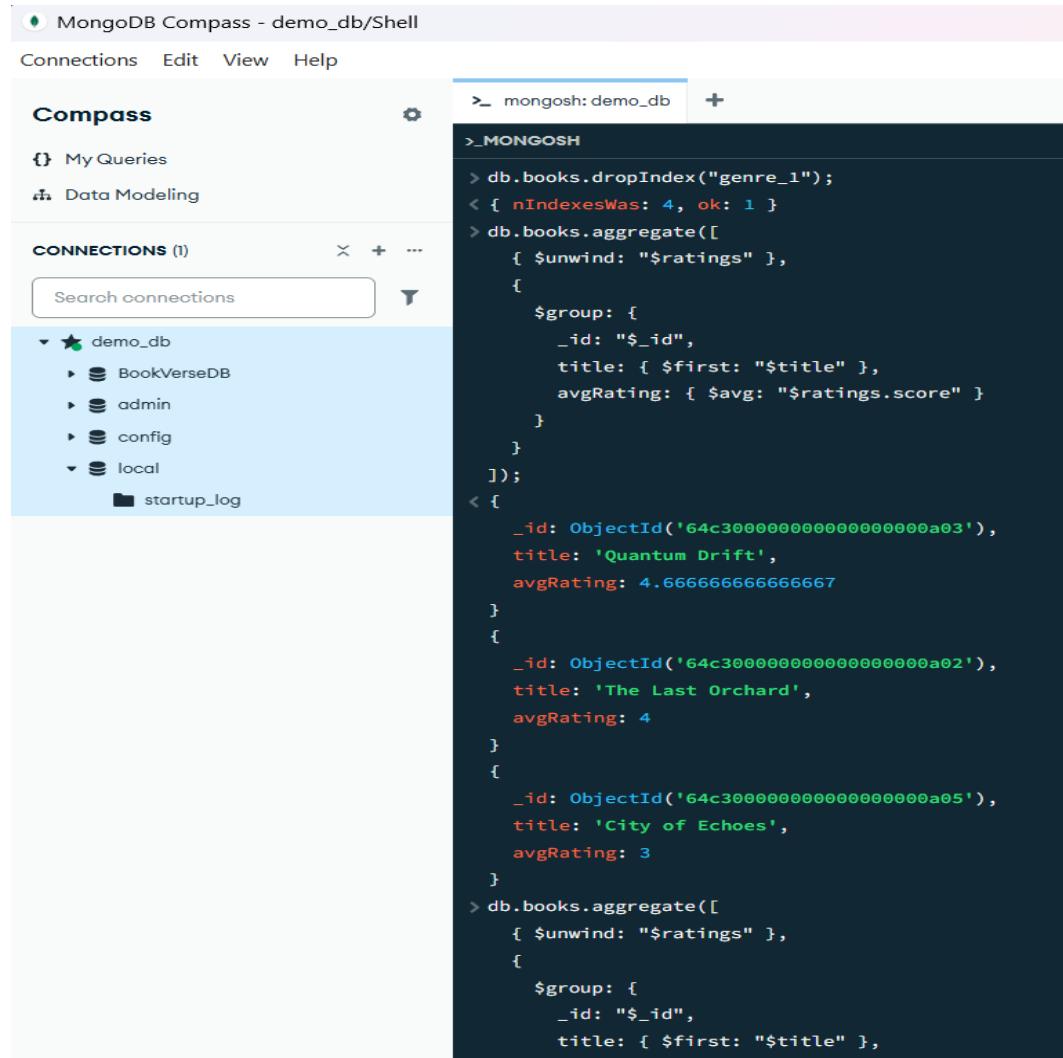

db.books.aggregate([
  {
    $group: {
      _id: "$authorId",
      totalBooks: { $sum: 1 }
    }
  },
  {
```

```

{ $match: { totalBooks: { $gt: 2 } } }
]);
db.books.aggregate([
{ $unwind: "$ratings" },
{
$group: {
_id: "$authorId",
totalPoints: { $sum: "$ratings.score" }
}
}
]);

```

## Output:-



The screenshot shows the MongoDB Compass interface with the title bar "MongoDB Compass - demo\_db/Shell". The left sidebar displays connections: "demo\_db" (selected), "BookVerseDB", "admin", "config", and "local". The "Connections" section has a search bar and a dropdown menu. The main panel shows the MONGOSH shell with the following aggregation pipeline:

```

> db.books.dropIndex("genre_1");
< { nIndexesWas: 4, ok: 1 }
> db.books.aggregate([
  { $unwind: "$ratings" },
  {
    $group: {
      _id: "$authorId",
      totalPoints: { $sum: "$ratings.score" }
    }
  }
]);

```

The results of the aggregation are displayed as an array of documents:

```

[{"_id": ObjectId('64c3000000000000000000a03'), "title": "Quantum Drift", "avgRating": 4.666666666666667}, {"_id": ObjectId('64c3000000000000000000a02'), "title": "The Last Orchard", "avgRating": 4}, {"_id": ObjectId('64c3000000000000000000a05'), "title": "City of Echoes", "avgRating": 3}]

```

MongoDB Compass - demo\_db/Shell

Connections Edit View Help

**Compass**

My Queries Data Modeling

**CONNECTIONS (1)**

Search connections

- demo\_db
  - BookVerseDB
  - admin
  - config
  - local
    - startup\_log

\_mongosh: demo\_db

```
>_ MONGOSH
> db.books.aggregate([
  {
    $group: {
      _id: "$genre",
      bookCount: { $sum: 1 }
    }
  }
]);
< [
  {
    _id: 'Mystery',
    bookCount: 1
  },
  {
    _id: 'Fantasy',
    bookCount: 2
  },
  {
    _id: 'Science Fiction',
    bookCount: 1
  }
]
> db.books.aggregate([
  {
    $group: {
      _id: "$authorId",
      totalBooks: { $sum: 1 }
    }
  },
  { $match: { totalBooks: { $gt: 2 } } }
]);
<
```

MongoDB Compass - demo\_db/Shell

Connections Edit View Help

**Compass**

My Queries Data Modeling

**CONNECTIONS (1)**

Search connections

- demo\_db
  - BookVerseDB
  - admin
  - config
  - local
    - startup\_log

\_mongosh: demo\_db

```
>_ MONGOSH
> db.books.aggregate([
  { $unwind: "$ratings" },
  {
    $group: {
      _id: "$_id",
      title: { $first: "$title" },
      avgRating: { $avg: "$ratings.score" }
    }
  },
  { $sort: { avgRating: -1 } },
  { $limit: 3 }
]);
< [
  {
    _id: ObjectId('64c3000000000000000000a03'),
    title: 'Quantum Drift',
    avgRating: 4.6666666666666667
  },
  {
    _id: ObjectId('64c3000000000000000000a02'),
    title: 'The Last Orchard',
    avgRating: 4
  },
  {
    _id: ObjectId('64c3000000000000000000a05'),
    title: 'City of Echoes',
    avgRating: 3
  }
]
> db.books.aggregate([
  {
    $group: {
```

The screenshot shows the MongoDB Compass interface. The top bar has tabs for 'Connections' (highlighted), 'Edit', 'View', and 'Help'. The left sidebar has sections for 'Compass', 'My Queries', 'Data Modeling', and 'CONNECTIONS (1)'. Under 'CONNECTIONS', there is a search bar and a list of databases: 'demo\_db' (selected, expanded to show 'BookVerseDB', 'admin', 'config', 'local', and 'startup\_log'), 'BookVerseDB', 'admin', 'config', and 'local'. The main right panel is titled 'mongosh: demo\_db' and contains a mongo shell session. The session shows the execution of two aggregate queries. The first query groups books by author ID and filters for authors with more than 2 books. The second query unwinds the ratings array for each book and groups them by author ID to calculate total points. The results show two authors: one with 7 total points and another with 14 total points.

```
>_ mongosh: demo_db +  
=>_ MONGOSH  
_id: 'Science Fiction',  
bookCount: 1  
}  
> db.books.aggregate([  
{  
$group: {  
_id: "$authorId",  
totalBooks: { $sum: 1 }  
}  
},  
{ $match: { totalBooks: { $gt: 2 } } }  
]);<  
> db.books.aggregate([  
{ $unwind: "$ratings" },  
{  
$group: {  
_id: "$authorId",  
totalPoints: { $sum: "$ratings.score" }  
}  
}  
]);<  
{  
_id: ObjectId('64a100000000000000000001'),  
totalPoints: 7  
}  
{  
_id: ObjectId('64a100000000000000000002'),  
totalPoints: 14  
}  
BookVerseDB >
```

## USER STORY 3 — MongoDB Atlas Connection

Code:-

```
// connect to Atlas
```

```
mongosh
```

```
"mongodb+srv://kiranpavanpamidi:Kiran%40321@cluster0.jnzx4vp.mongodb.net/"
```

```
//database
```

```
use BookVerseDB
```

```
show collections
```

```
db.books.find()
```

## Output:-

```
mongosh mongodb+srv:// Kiran%40321@cluster0.jnxz4vp.mongodb.net/
```

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

```
PS C:\Users\kiran> mongosh "mongodb+srv://kirantanpavanpamidi:Kiran%40321@cluster0.jnxz4vp.mongodb.net/"
```

Current Mongosh Log ID: 691852b71f45bbd24963b111  
Connecting to: <mongodb+srv://<credentials>@cluster0.jnxz4vp.mongodb.net/?appName=mongosh+2.5.9>  
Using MongoDB: 8.0.15  
Using Mongosh: 2.5.9

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

```
Atlas atlas-4n4l1r-shard-0 [primary] test> use BookVerseDB
switched to db BookVerseDB
Atlas atlas-4n4l1r-shard-0 [primary] BookVerseDB> show collections
authors
books
users
Atlas atlas-4n4l1r-shard-0 [primary] BookVerseDB> db.books.find()
[
  {
    _id: ObjectId('64c3000000000000000000a02'),
    title: 'The Last Orchard',
    genre: 'Fantasy',
    publicationYear: 2012,
    authorId: ObjectId('64a1000000000000000000001'),
    ratings: [ { user: 'omar@example.com', score: 4, comment: 'Emotional.' } ],
    __v: 0
  },
  {
    _id: ObjectId('64c3000000000000000000a03'),
    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, comment: 'A must-read.' },
      {
        user: 'sara@example.com',
        score: 4,
        comment: 'Complex but good.'
      }
    ],
    __v: 0
  },
  {
    _id: ObjectId('64c3000000000000000000a04'),
    title: 'Wind of the Wild',
    genre: 'Fantasy',
    publicationYear: 2019,
    authorId: ObjectId('64a100000000000000000003'),
    ratings: [],
    __v: 0
  },
  {
    _id: ObjectId('64c3000000000000000000a05'),
    title: 'City of Echoes',
    genre: 'Mystery',
    publicationYear: 2016,
    authorId: ObjectId('64a100000000000000000001'),
    ratings: [ { user: 'sara@example.com', score: 3, comment: 'Okay.' } ],
    __v: 0
  }
]
Atlas atlas-4n4l1r-shard-0 [primary] BookVerseDB> |
```