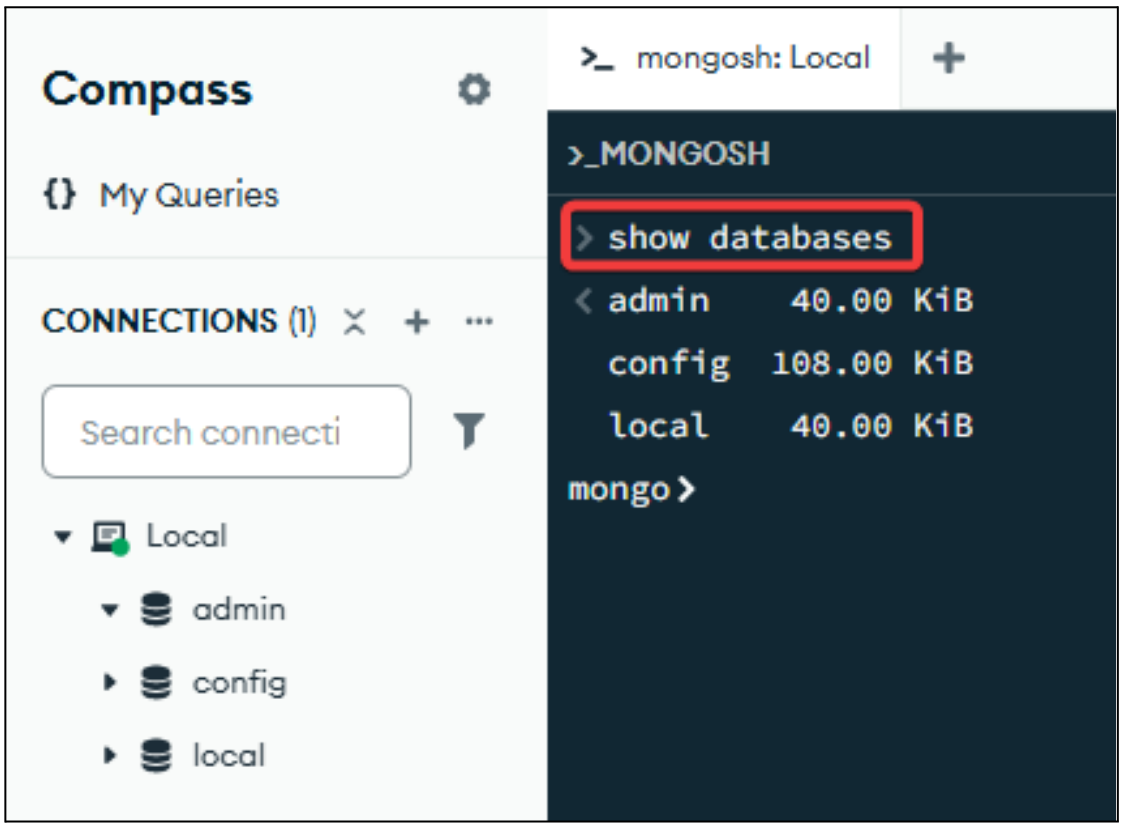
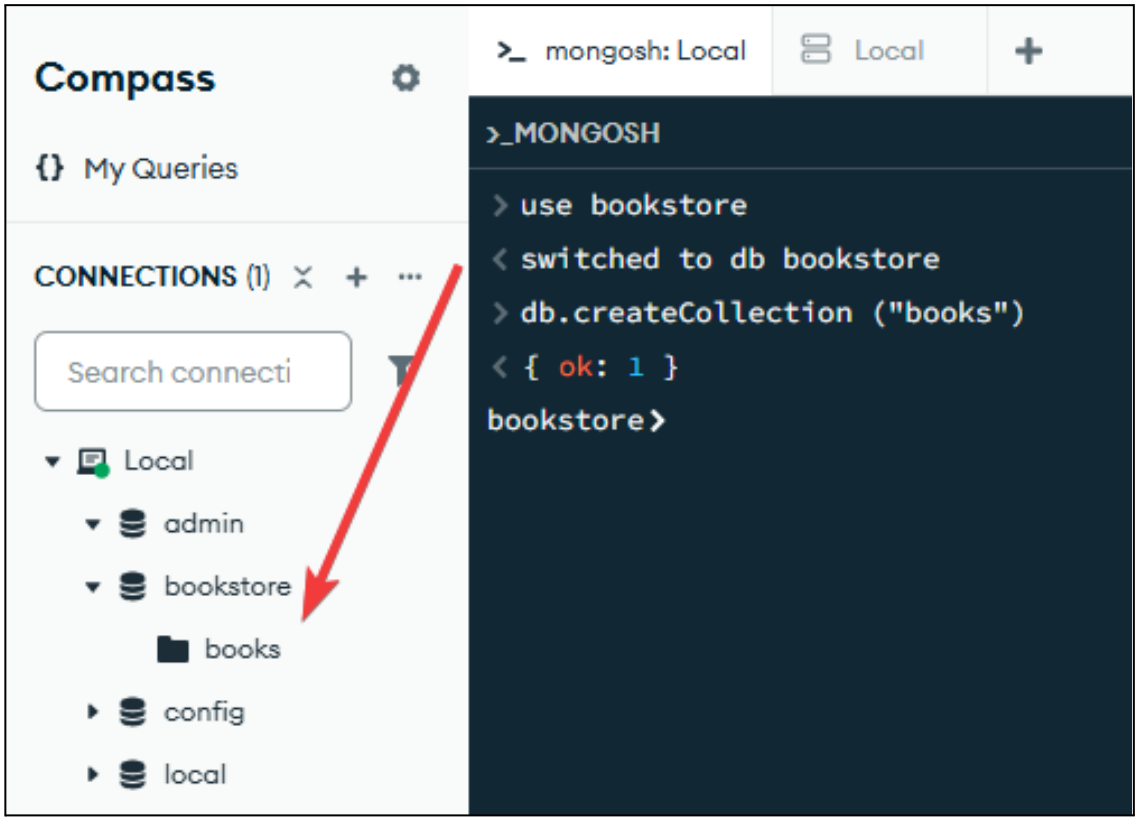
Database & Collection Setup

1. Show all databases.



2. Create a database ‘bookstore’ and the ‘books’ collection.

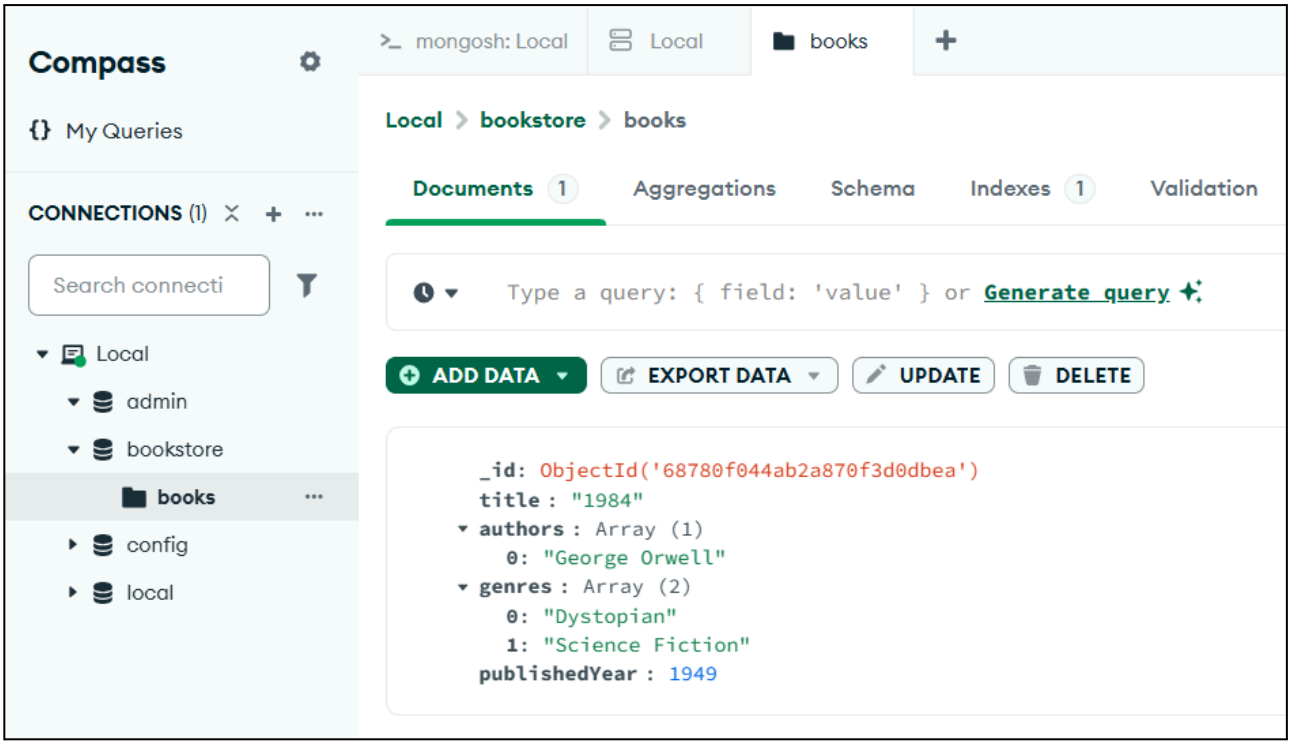


Insert Operations

3. Insert one document into the collection and verify it has been added.

```
> db.books.insertOne({title: "1984", authors: ["George Orwell"], genres: ["Dystopian", "Science Fiction"],publishedYear: 1949})
< {
    acknowledged: true,
    insertedId: ObjectId('68780f044ab2a870f3d0dbea')
  }
> db.books.find()
< {
    _id: ObjectId('68780f044ab2a870f3d0dbea'),
    title: '1984',
    authors: [
      'George Orwell'
    ],
    genres: [
      'Dystopian',
      'Science Fiction'
    ],
    publishedYear: 1949
  }
```

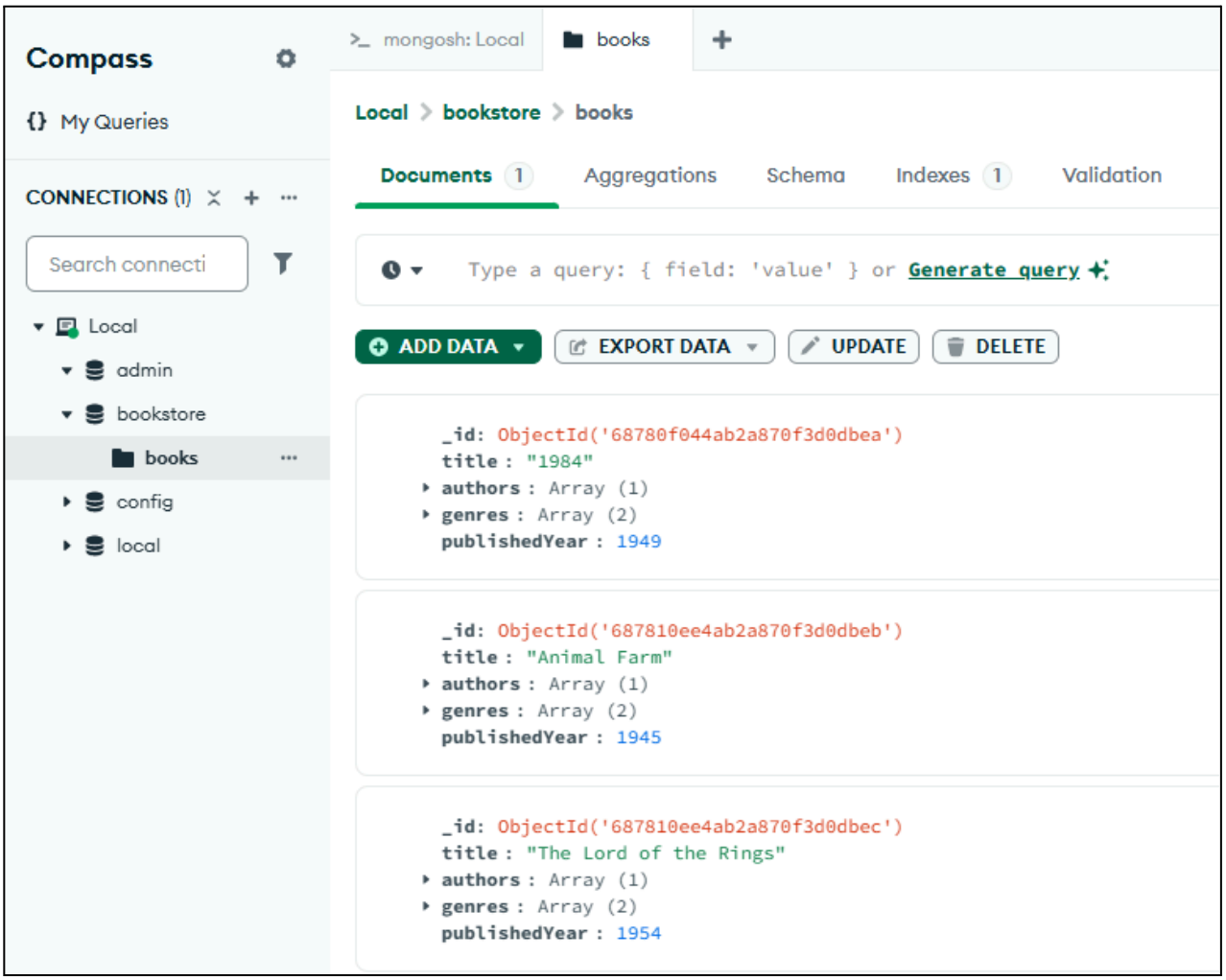
Check that the document was successfully added:



4. Insert many documents and check that they have been added.

```
>_MONGOSH
> db.books.insertMany([
    {title: "Animal Farm", authors: ["George Orwell"], genres: [ "Allegory", "Fiction"], publishedYear: 1945},
    {title: "The Lord of the Rings", authors: ["J.R.R. Tolkien"], genres: ["Fantasy", "Adventure"],publishedYear: 1954 }
    ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('687810ee4ab2a870f3d0dbeb'),
      '1': ObjectId('687810ee4ab2a870f3d0dbec')
    }
  }
```

Check that the document was successfully added:



5. Insert a document with nested fields into the users collection

```
> db.books.insertOne({ title: "The Great Gatsby", authors: ["F. Scott Fitzgerald"],  genres: ["Classic", "Tragedy"],
    publishedYear: 1925, reviews: [{name: "Oliver", body: "It is a great book!"}, {name: "Olivia", body: "Not my cup of tea"} ]})
< {
  acknowledged: true,
  insertedId: ObjectId('68790d5c4ab2a870f3d0dbed')
}
```

Check that the document was successfully added:

```
> db.books.find().sort ({_id: -1}).limit(1)
< {
    _id: ObjectId('68790d5c4ab2a870f3d0dbed'),
    title: 'The Great Gatsby',
    authors: [
      'F. Scott Fitzgerald'
    ],
    genres: [
      'Classic',
      'Tragedy'
    ],
    publishedYear: 1925,
    reviews: [
      {
        name: 'Oliver',
        body: 'It is a great book!'
      },
      {
        name: 'Olivia',
        body: 'Not my cup of tea'
      }
    ]
  }
```

Basic Queries

6. Search for documents by the author George Orwell.

```
> db.books.find({authors: "George Orwell"})
< {
    _id: ObjectId('68780f044ab2a870f3d0dbea'),
    title: '1984',
    authors: [
      'George Orwell'
    ],
    genres: [
      'Dystopian',
      'Science Fiction'
    ],
    publishedYear: 1949
  }
  {
    _id: ObjectId('687810ee4ab2a870f3d0dbeb'),
    title: 'Animal Farm',
    authors: [
      'George Orwell'
    ],
    genres: [
      'Allegory',
      'Fiction'
    ],
    publishedYear: 1945
  }
```

7. Find documents matching multiple conditions.

```
> db.books.find({authors: "George Orwell", genres: "Fiction"})
< {
    _id: ObjectId('687810ee4ab2a870f3d0dbeb'),
    title: 'Animal Farm',
    authors: [
      'George Orwell'
    ],
    genres: [
      'Allegory',
      'Fiction'
    ],
    publishedYear: 1945
  }
```

8. Find documents by condition and select specific fields.

```
> db.books.find({authors:"George Orwell"}, {title:1, authors:1})
< {
    _id: ObjectId('68780f044ab2a870f3d0dbea'),
    title: '1984',
    authors: [
      'George Orwell'
    ]
  }
  {
    _id: ObjectId('687810ee4ab2a870f3d0dbeb'),
    title: 'Animal Farm',
    authors: [
      'George Orwell'
    ]
  }
```

9. Select specific fields from all documents.

```
> db.books.find({}, {_id:0,title:1, authors:1})
< {
    title: '1984',
    authors: [
      'George Orwell'
    ]
  }
  {
    title: 'Animal Farm',
    authors: [
      'George Orwell'
    ]
  }
  {
    title: 'The Lord of the Rings',
    authors: [
      'J.R.R. Tolkien'
    ]
  }
```

10. Find a Document Using the **findOne** method in MongoDB.

```
> db.books.findOne({authors: "George Orwell"})
< {
    _id: ObjectId('68780f044ab2a870f3d0dbea'),
    title: '1984',
    authors: [
      'George Orwell'
    ],
    genres: [
      'Dystopian',
      'Science Fiction'
    ],
    publishedYear: 1949
  }
```

Count, Limit, Sort

11. Get the total number of documents in the "books" collection.

```
> db.books.find().count()
< 3
> db.books.find({authors:"George Orwell"}).count()
< 2
```

12. Get the first document from the "books" collection.

```
>_MONGOSH
> db.books.find().limit (1)
< {
    _id: ObjectId('68780f044ab2a870f3d0dbea'),
    title: '1984',
    authors: [
      'George Orwell'
    ],
    genres: [
      'Dystopian',
      'Science Fiction'
    ],
    publishedYear: 1949
  }
```

13. Sort documents in the "books" collection by title in descending order and genre in ascending order

```
>_MONGOSH
> db.books.find().sort({title:-1, genre:1})
< {
    _id: ObjectId('687810ee4ab2a870f3d0dbec'),
    title: 'The Lord of the Rings',
    authors: [
      'J.R.R. Tolkien'
    ],
    genres: [
      'Fantasy',
      'Adventure'
    ],
    publishedYear: 1954
  }
  {
    _id: ObjectId('687810ee4ab2a870f3d0dbeb'),
    title: 'Animal Farm',
    authors: [
      'George Orwell'
    ],
    genres: [
      'Allegory',
      'Fiction'
    ],
    publishedYear: 1945
  }
  {
    _id: ObjectId('68780f044ab2a870f3d0dbea'),
    title: '1984',
    authors: [
      'George Orwell'
    ],
    genres: [
      'Dystopian',
      'Science Fiction'
    ],
    publishedYear: 1949
  }
```

Query operators

($gt, $gte, $lt, $or, $and, $in, $nin)

14. Find all books published in or after 1949.

```
> db.books.find({publishedYear: {$gte:1949}})
< {
    _id: ObjectId('68780f044ab2a870f3d0dbea'),
    title: '1984',
    authors: [
      'George Orwell'
    ],
    genres: [
      'Dystopian',
      'Science Fiction'
    ],
    publishedYear: 1949
  }
  {
    _id: ObjectId('687810ee4ab2a870f3d0dbec'),
    title: 'The Lord of the Rings',
    authors: [
      'J.R.R. Tolkien'
    ],
    genres: [
      'Fantasy',
      'Adventure'
    ],
    publishedYear: 1954
  }
```

15. Find all books published before 1945.

```
> db.books.find({publishedYear: {$lt: 1945}})
< {
    _id: ObjectId('68790d5c4ab2a870f3d0dbed'),
    title: 'The Great Gatsby',
    authors: [
      'F. Scott Fitzgerald'
    ],
    genres: [
      'Classic',
      'Tragedy'
    ],
    publishedYear: 1925,
    reviews: [
      {
        name: 'Oliver',
        body: 'It is a great book!'
      },
      {
        name: 'Olivia',
        body: 'Not my cup of tea'
      }
    ]
  }
```

16. Find books written in the Adventure genre or published in 1945.

```
> db.books.find({$or: [{publishedYear:1945}, {genres: "Adventure"}] })
< {
    _id: ObjectId('687810ee4ab2a870f3d0dbeb'),
    title: 'Animal Farm',
    authors: [
      'George Orwell'
    ],
    genres: [
      'Allegory',
      'Fiction'
    ],
    publishedYear: 1945
  }
  {
    _id: ObjectId('687810ee4ab2a870f3d0dbec'),
    title: 'The Lord of the Rings',
    authors: [
      'J.R.R. Tolkien'
    ],
    genres: [
      'Fantasy',
      'Adventure'
    ],
    publishedYear: 1954
  }
```

17. Find books written by "George Orwell" and published after 1945.

```
> db.books.find({$and:[ {publishedYear:{$gt:1945}},{authors:"George Orwell"} ]})
< {
    _id: ObjectId('68780f044ab2a870f3d0dbea'),
    title: '1984',
    authors: [
      'George Orwell'
    ],
    genres: [
      'Dystopian',
      'Science Fiction'
    ],
    publishedYear: 1949
  }
```

18.Find books published in either 1945,1946,1947,1948, or 1949.

```
> db.books.find({publishedYear: {$in: [1945,1946,1947,1948,1949]}})
< {
    _id: ObjectId('68780f044ab2a870f3d0dbea'),
    title: '1984',
    authors: [
      'George Orwell'
    ],
    genres: [
      'Dystopian',
      'Science Fiction'
    ],
    publishedYear: 1949
  }
  {
    _id: ObjectId('687810ee4ab2a870f3d0dbeb'),
    title: 'Animal Farm',
    authors: [
      'George Orwell'
    ],
    genres: [
      'Allegory',
      'Fiction'
    ],
    publishedYear: 1945
  }
```

19. Find books not written by "George Orwell" or "F. Scott Fitzgerald".

```
> db.books.find({ authors: {$nin: ["George Orwell","F. Scott Fitzgerald"] }    })
< {
    _id: ObjectId('687810ee4ab2a870f3d0dbec'),
    title: 'The Lord of the Rings',
    authors: [
      'J.R.R. Tolkien'
    ],
    genres: [
      'Fantasy',
      'Adventure'
    ],
    publishedYear: 1954
  }
```

20. Find books with reviewer name "Oliver"

```
> db.books.find({"reviews.name":"Oliver"})
< {
    _id: ObjectId('68790d5c4ab2a870f3d0dbed'),
    title: 'The Great Gatsby',
    authors: [
      'F. Scott Fitzgerald'
    ],
    genres: [
      'Classic',
      'Tragedy'
    ],
    publishedYear: 1925,
    reviews: [
      {
        name: 'Oliver',
        body: 'It is a great book!'
      },
      {
        name: 'Olivia',
        body: 'Not my cup of tea'
      }
    ]
  }
```

Update & Delete

21.Update the title of a book

```
> db.books.updateOne({_id: ObjectId("68790d5c4ab2a870f3d0dbed")}, {$set: {title: "The Great Gatsby."} } )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.books.find()
< {
    _id: ObjectId('687810ee4ab2a870f3d0dbec'),
    title: 'The Lord of the Rings',
    authors: [
      'J.R.R. Tolkien'
    ],
    genres: [
      'Fantasy',
      'Adventure'
    ],
    publishedYear: 1954
  }
  {
    _id: ObjectId('68790d5c4ab2a870f3d0dbed'),
    title: 'The Great Gatsby.',
    authors: [
```

22. Delete one book by id

```
> db.books.deleteOne({_id: ObjectId('687810ee4ab2a870f3d0dbeb')})
< {
    acknowledged: true,
    deletedCount: 1
  }
```

23. Delete all books written by George Orwell

```
> db.books.deleteMany({authors:"George Orwell"} )
< {
    acknowledged: true,
    deletedCount: 1
  }
```