

EXPERIMENT 6

Name:- Omkar Kore

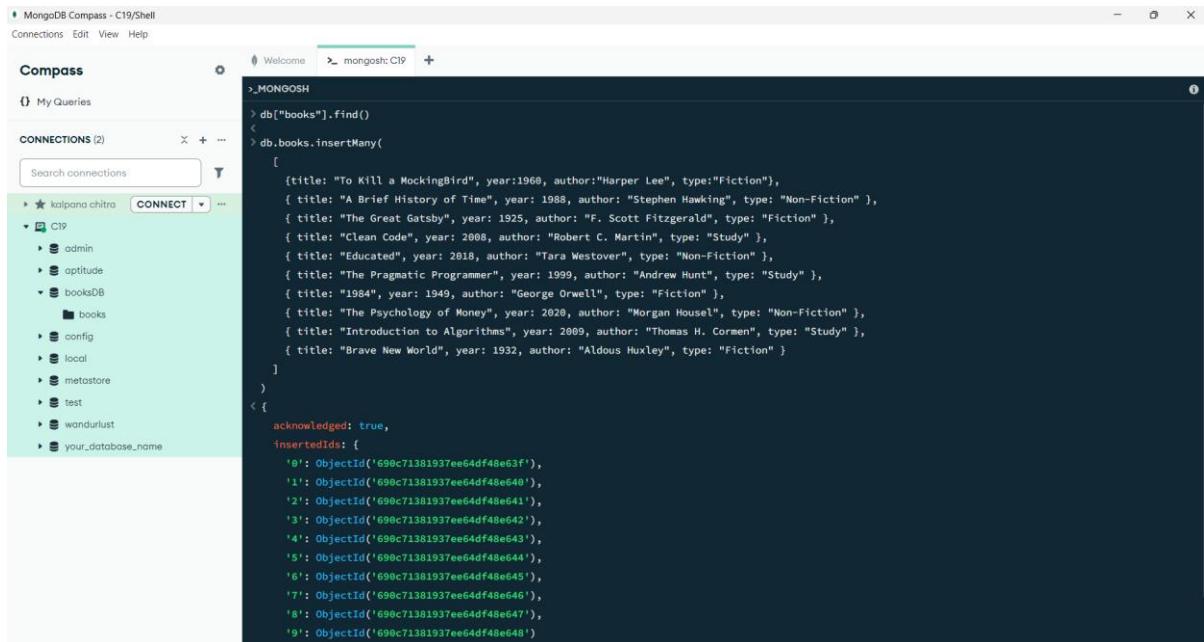
Class: Btech C

Roll Number:- BTC19

Batch: C1

MongoDB Queries

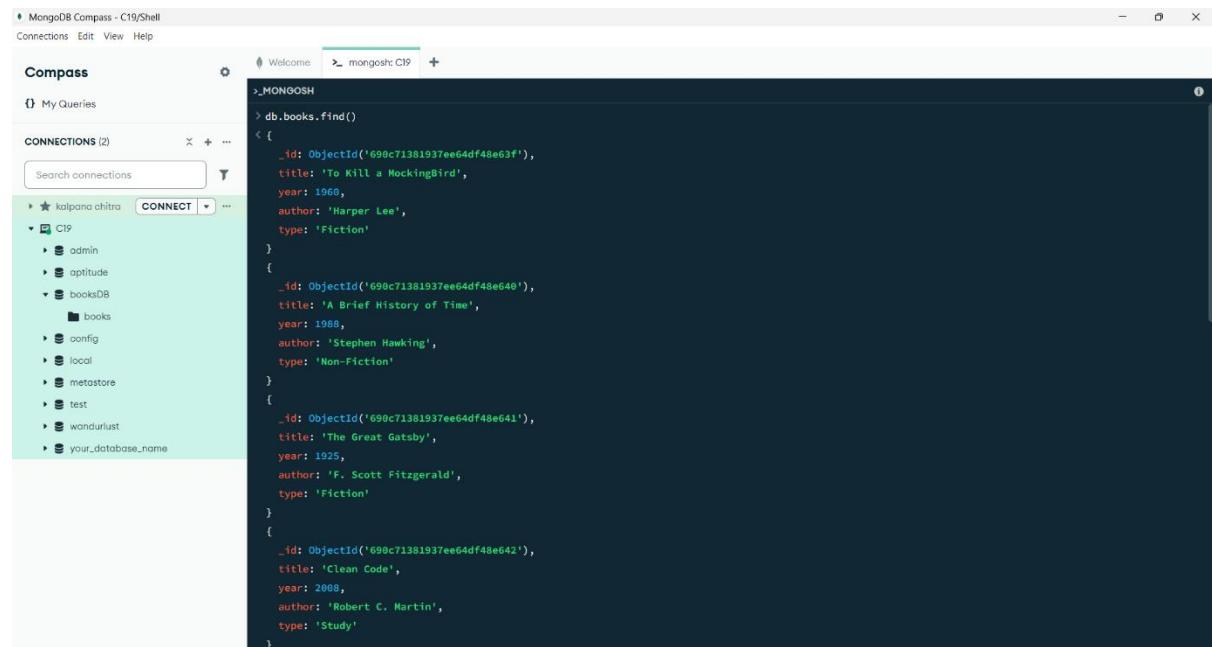
1. Inserting Books



```
>_MONGOSH
> db["books"].find()
<
> db.books.insertMany([
  {
    title: "To Kill a MockingBird", year:1960, author:"Harper Lee", type:"Fiction",
    title: "A Brief History of Time", year: 1988, author: "Stephen Hawking", type: "Non-Fiction" ,
    title: "The Great Gatsby", year: 1925, author: "F. Scott Fitzgerald", type: "Fiction" ,
    title: "Clean Code", year: 2008, author: "Robert C. Martin", type: "Study" ,
    title: "Educated", year: 2018, author: "Tara Westover", type: "Non-Fiction" ,
    title: "The Pragmatic Programmer", year: 1999, author: "Andrew Hunt", type: "Study" ,
    title: "1984", year: 1949, author: "George Orwell", type: "Fiction" ,
    title: "The Psychology of Money", year: 2020, author: "Morgan Housel", type: "Non-Fiction" ,
    title: "Introduction to Algorithms", year: 2009, author: "Thomas H. Cormen", type: "Study" ,
    title: "Brave New World", year: 1932, author: "Aldous Huxley", type: "Fiction" 
  ]
)
< {
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('690c71381937ee64df48e63f'),
    '1': ObjectId('690c71381937ee64df48e640'),
    '2': ObjectId('690c71381937ee64df48e641'),
    '3': ObjectId('690c71381937ee64df48e642'),
    '4': ObjectId('690c71381937ee64df48e643'),
    '5': ObjectId('690c71381937ee64df48e644'),
    '6': ObjectId('690c71381937ee64df48e645'),
    '7': ObjectId('690c71381937ee64df48e646'),
    '8': ObjectId('690c71381937ee64df48e647'),
    '9': ObjectId('690c71381937ee64df48e648')
  ]
}
```

```
db.books.insertMany(
[
  {
    title: "To Kill a MockingBird", year:1960, author:"Harper Lee", type:"Fiction",
    title: "A Brief History of Time", year: 1988, author: "Stephen Hawking", type: "Non-Fiction" ,
    title: "The Great Gatsby", year: 1925, author: "F. Scott Fitzgerald", type: "Fiction" ,
    title: "Clean Code", year: 2008, author: "Robert C. Martin", type: "Study" ,
    title: "Educated", year: 2018, author: "Tara Westover", type: "Non-Fiction" ,
    title: "The Pragmatic Programmer", year: 1999, author: "Andrew Hunt", type: "Study" ,
    title: "1984", year: 1949, author: "George Orwell", type: "Fiction" ,
    title: "The Psychology of Money", year: 2020, author: "Morgan Housel", type: "Non-Fiction" ,
    title: "Introduction to Algorithms", year: 2009, author: "Thomas H. Cormen", type: "Study" ,
    title: "Brave New World", year: 1932, author: "Aldous Huxley", type: "Fiction" 
  ]
)
```

2. Show All Books

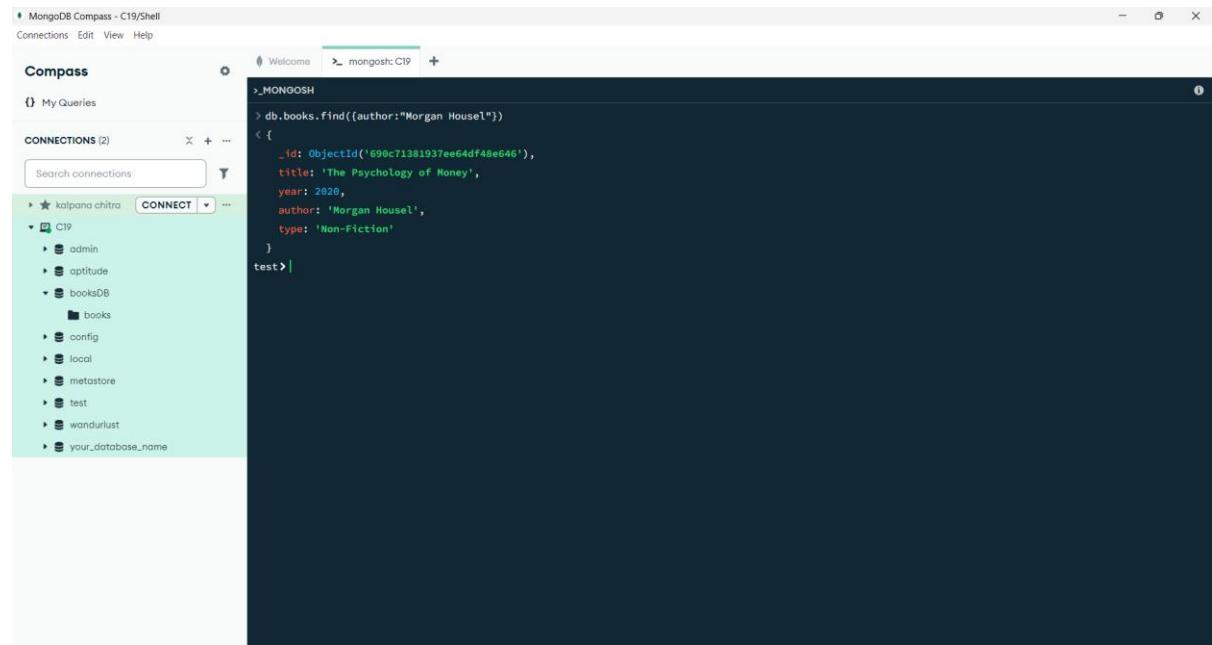


The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists two connections: 'kalpana chitra' and 'C19'. The 'C19' connection is selected, showing its database structure. The main pane displays the results of a MongoDB shell command: `> db.books.find()`. The output shows four documents from the 'books' collection:

```
> db.books.find()
< [
  {
    "_id": ObjectId('690c71381937ee64df48e63f'),
    "title": "To Kill a MockingBird",
    "year": 1960,
    "author": "Harper Lee",
    "type": "Fiction"
  },
  {
    "_id": ObjectId('690c71381937ee64df48e640'),
    "title": "A Brief History of Time",
    "year": 1988,
    "author": "Stephen Hawking",
    "type": "Non-Fiction"
  },
  {
    "_id": ObjectId('690c71381937ee64df48e641'),
    "title": "The Great Gatsby",
    "year": 1925,
    "author": "F. Scott Fitzgerald",
    "type": "Fiction"
  },
  {
    "_id": ObjectId('690c71381937ee64df48e642'),
    "title": "Clean Code",
    "year": 2008,
    "author": "Robert C. Martin",
    "type": "Study"
  }
]
```

`db.books.find()`

3. Find books by a specific author

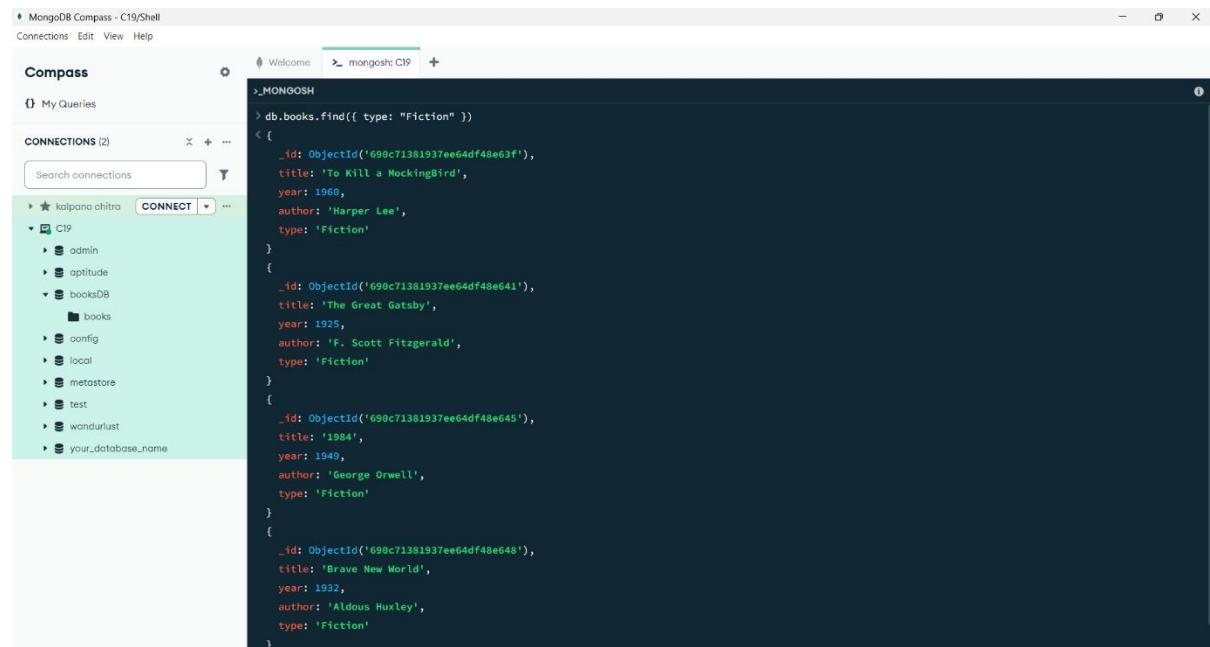


The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists two connections: 'kalpana chitra' and 'C19'. The 'C19' connection is selected, showing its database structure. The main pane displays the results of a MongoDB shell command: `> db.books.find({author:"Morgan Housel"})`. The output shows one document from the 'books' collection:

```
> db.books.find({author:"Morgan Housel"})
< [
  {
    "_id": ObjectId('690c71381937ee64df48e646'),
    "title": "The Psychology of Money",
    "year": 2020,
    "author": "Morgan Housel",
    "type": "Non-Fiction"
  }
]
```

`db.books.find({author:"Morgan Housel"})`

4. Find All Fiction Books

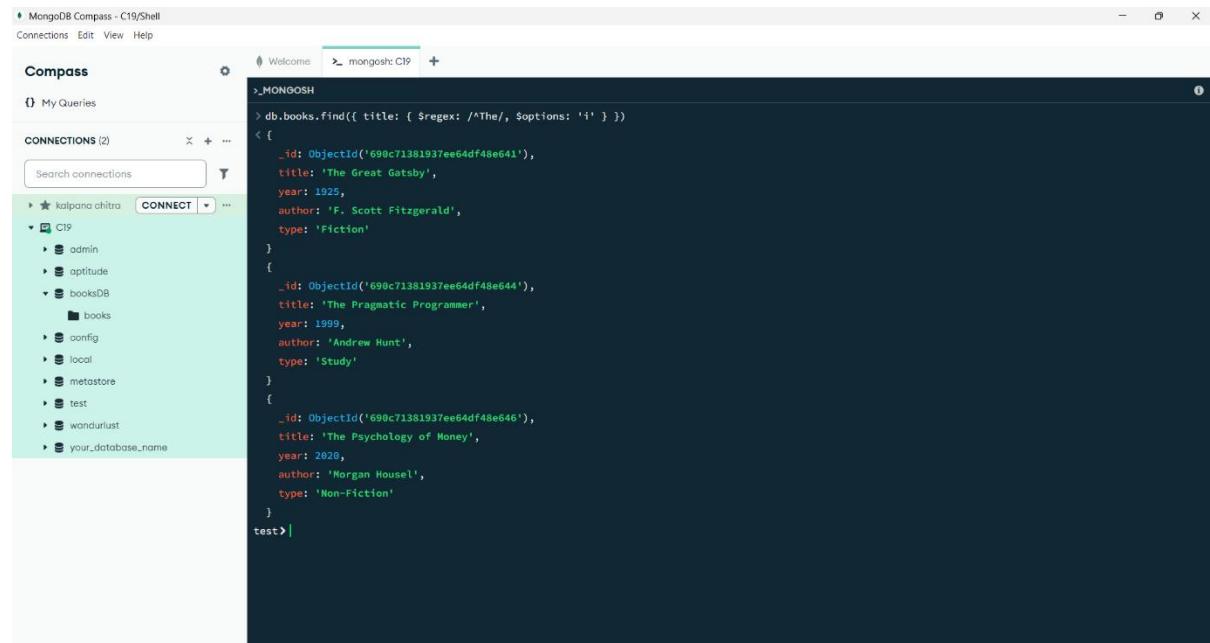


The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists two connections: 'kalpana chitra' and 'C19'. The 'C19' connection is selected, showing its database structure. The main panel displays the results of a MongoDB shell command: `> db.books.find({ type: "Fiction" })`. The output shows five documents representing fiction books:

```
> db.books.find({ type: "Fiction" })
< [
  {
    _id: ObjectId('690c71381937ee64df48e63f'),
    title: 'To Kill a MockingBird',
    year: 1960,
    author: 'Harper Lee',
    type: 'Fiction'
  },
  {
    _id: ObjectId('690c71381937ee64df48e641'),
    title: 'The Great Gatsby',
    year: 1925,
    author: 'F. Scott Fitzgerald',
    type: 'Fiction'
  },
  {
    _id: ObjectId('690c71381937ee64df48e645'),
    title: '1984',
    year: 1949,
    author: 'George Orwell',
    type: 'Fiction'
  },
  {
    _id: ObjectId('690c71381937ee64df48e648'),
    title: 'Brave New World',
    year: 1932,
    author: 'Aldous Huxley',
    type: 'Fiction'
  }
]
```

```
db.books.find({ type: "Fiction" })
```

5. Find all books whose title starts with “THE”

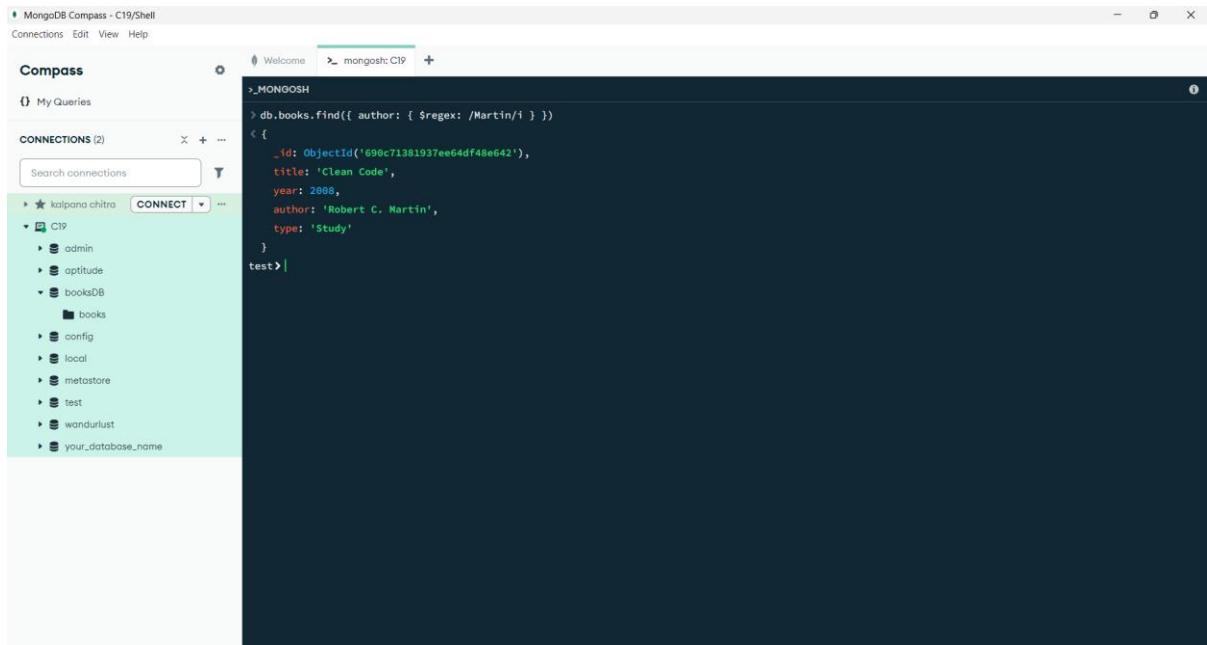


The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists two connections: 'kalpana chitra' and 'C19'. The 'C19' connection is selected, showing its database structure. The main panel displays the results of a MongoDB shell command: `> db.books.find({ title: { $regex: /^The/, $options: 'i' } })`. The output shows three documents whose titles start with 'The':

```
> db.books.find({ title: { $regex: /^The/, $options: 'i' } })
< [
  {
    _id: ObjectId('690c71381937ee64df48e641'),
    title: 'The Great Gatsby',
    year: 1925,
    author: 'F. Scott Fitzgerald',
    type: 'Fiction'
  },
  {
    _id: ObjectId('690c71381937ee64df48e644'),
    title: 'The Pragmatic Programmer',
    year: 1999,
    author: 'Andrew Hunt',
    type: 'Study'
  },
  {
    _id: ObjectId('690c71381937ee64df48e646'),
    title: 'The Psychology of Money',
    year: 2020,
    author: 'Morgan Housel',
    type: 'Non-Fiction'
  }
]
```

```
db.books.find({ title: { $regex: /^The/, $options: 'i' } })
```

6. Find all authors whose name contains “Martin”

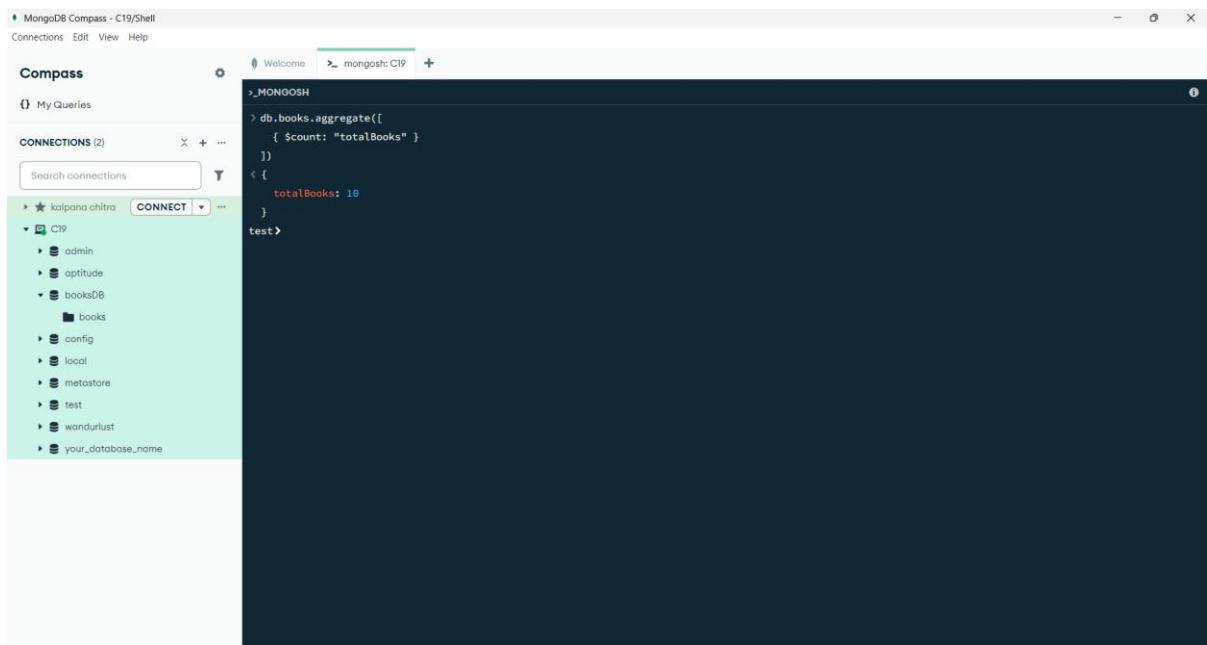


The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists two connections: 'kalpana chitra' and 'C19'. The 'C19' connection is selected, showing its database structure. The right panel is the mongo shell interface with the command and its output:

```
> db.books.find({ author: { $regex: /Martin/i } })
< [
  {
    _id: ObjectId('690c71381937ee64df48e642'),
    title: 'Clean Code',
    year: 2008,
    author: 'Robert C. Martin',
    type: 'Study'
  }
]
test>
```

```
db.books.find({ author: { $regex: /Martin/i } })
```

7. Count Total Books

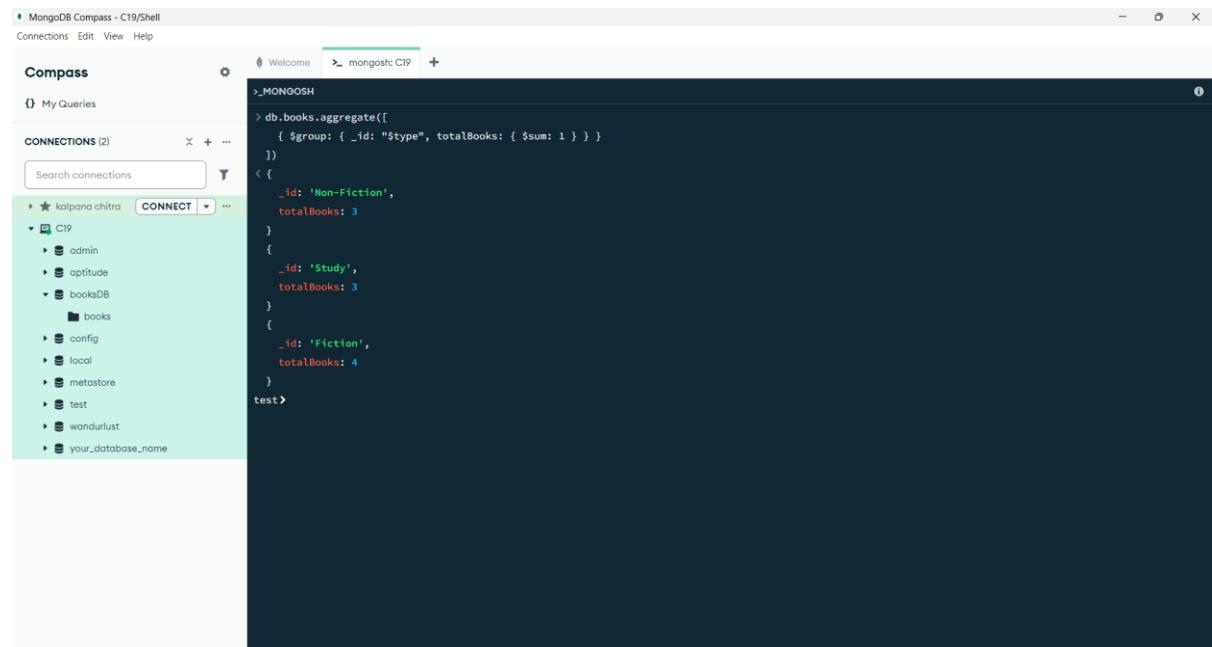


The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists two connections: 'kalpana chitra' and 'C19'. The 'C19' connection is selected, showing its database structure. The right panel is the mongo shell interface with the command and its output:

```
> db.books.aggregate([
  { $count: "totalBooks" }
])
< [
  {
    totalBooks: 10
  }
]
test>
```

```
db.books.aggregate([
  { $count: "totalBooks" }
])
```

8. Count books by type

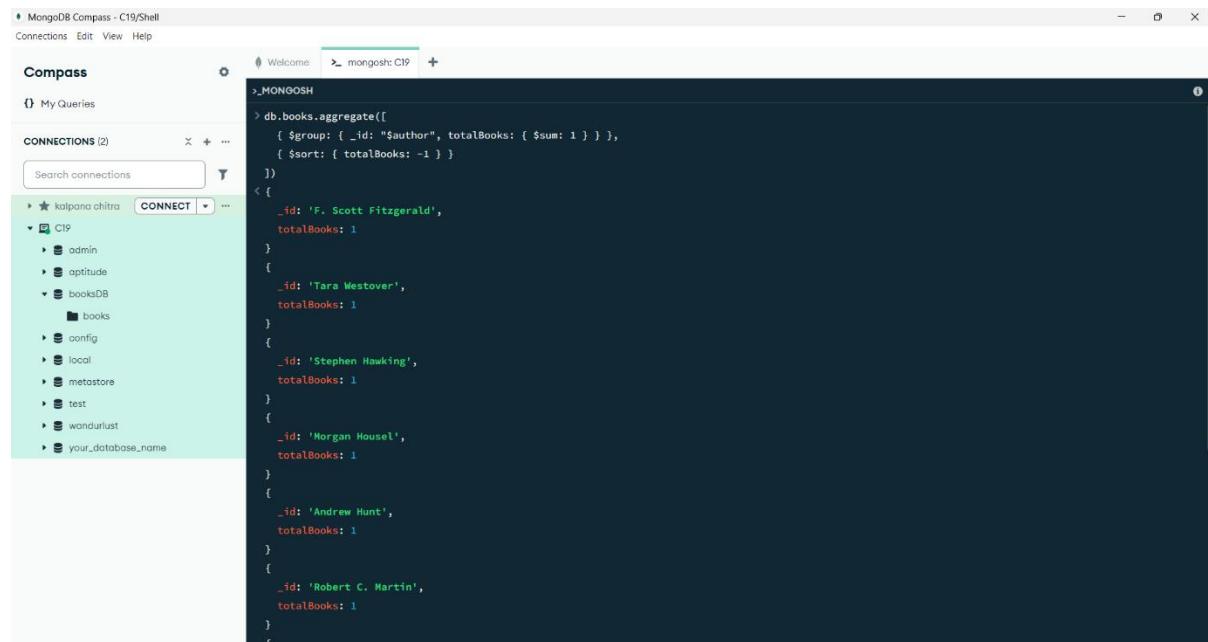


The screenshot shows the MongoDB Compass interface. On the left, the 'Compass' sidebar displays a list of connections, with 'C19' selected. The main panel shows the MongoDB shell command line. The command entered is:

```
>_MONGOSH  
> db.books.aggregate([  
  { $group: { _id: "$type", totalBooks: { $sum: 1 } } }  
)  
< [  
  {  
    _id: 'Non-Fiction',  
    totalBooks: 3  
  },  
  {  
    _id: 'Study',  
    totalBooks: 3  
  },  
  {  
    _id: 'Fiction',  
    totalBooks: 4  
  }  
test>
```

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

9. Count books per author

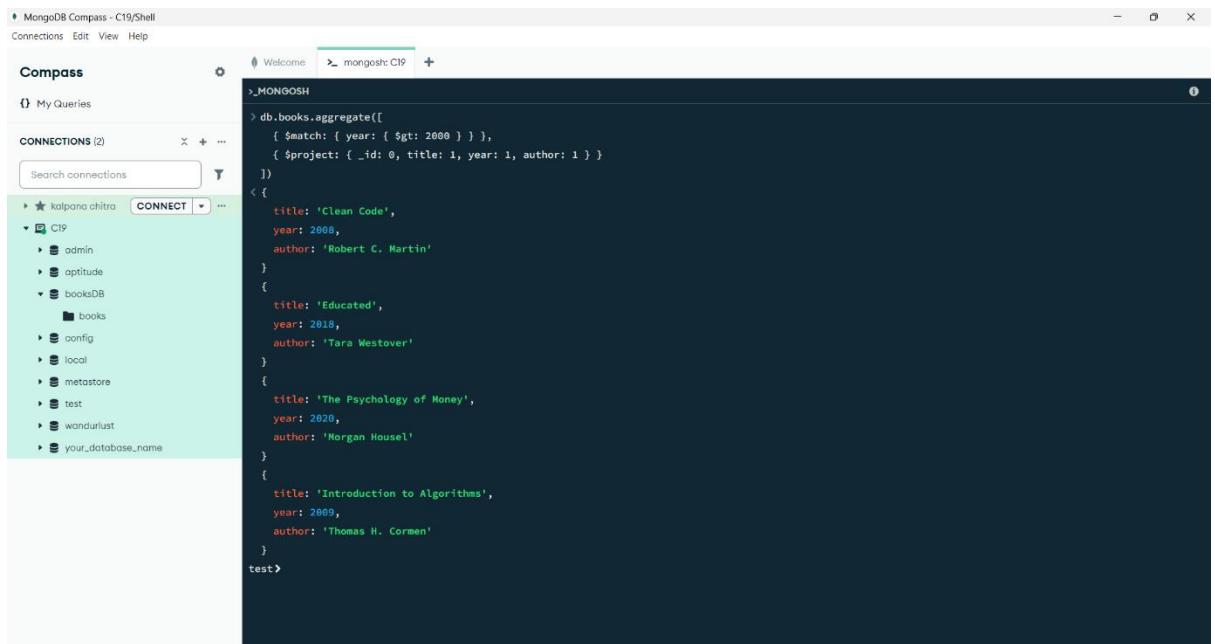


The screenshot shows the MongoDB Compass interface. On the left, the 'Compass' sidebar displays a list of connections, with 'C19' selected. The main panel shows the MongoDB shell command line. The command entered is:

```
>_MONGOSH  
> db.books.aggregate([  
  { $group: { _id: "$author", totalBooks: { $sum: 1 } } },  
  { $sort: { totalBooks: -1 } }  
)  
< [  
  {  
    _id: 'F. Scott Fitzgerald',  
    totalBooks: 1  
  },  
  {  
    _id: 'Tara Westover',  
    totalBooks: 1  
  },  
  {  
    _id: 'Stephen Hawking',  
    totalBooks: 1  
  },  
  {  
    _id: 'Morgan Housel',  
    totalBooks: 1  
  },  
  {  
    _id: 'Andrew Hunt',  
    totalBooks: 1  
  },  
  {  
    _id: 'Robert C. Martin',  
    totalBooks: 1  
  }  
test>
```

```
db.books.aggregate([  
  { $group: { _id: "$author", totalBooks: { $sum: 1 } } },  
  { $sort: { totalBooks: -1 } }  
)
```

10. Books published after year 2000

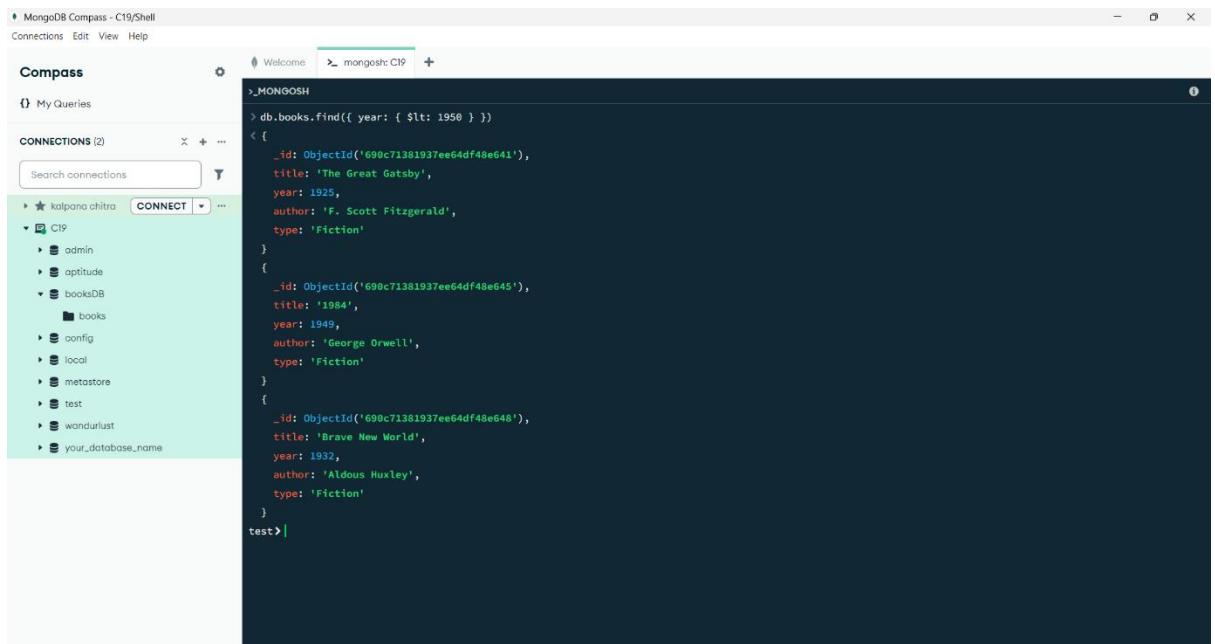


The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists 'C19' as the selected connection. The main area displays the MONGOSH shell with the following command:

```
> db.books.aggregate([
  { $match: { year: { $gt: 2000 } } },
  { $project: { _id: 0, title: 1, year: 1, author: 1 } }
])
< [
  {
    title: 'Clean Code',
    year: 2008,
    author: 'Robert C. Martin'
  },
  {
    title: 'Educated',
    year: 2018,
    author: 'Tara Westover'
  },
  {
    title: 'The Psychology of Money',
    year: 2020,
    author: 'Morgan Housel'
  },
  {
    title: 'Introduction to Algorithms',
    year: 2009,
    author: 'Thomas H. Cormen'
  }
]
test>
```

```
db.books.aggregate([
  { $match: { year: { $gt: 2000 } } },
  { $project: { _id: 0, title: 1, year: 1, author: 1 } }
])
```

11. Find books published before 1950



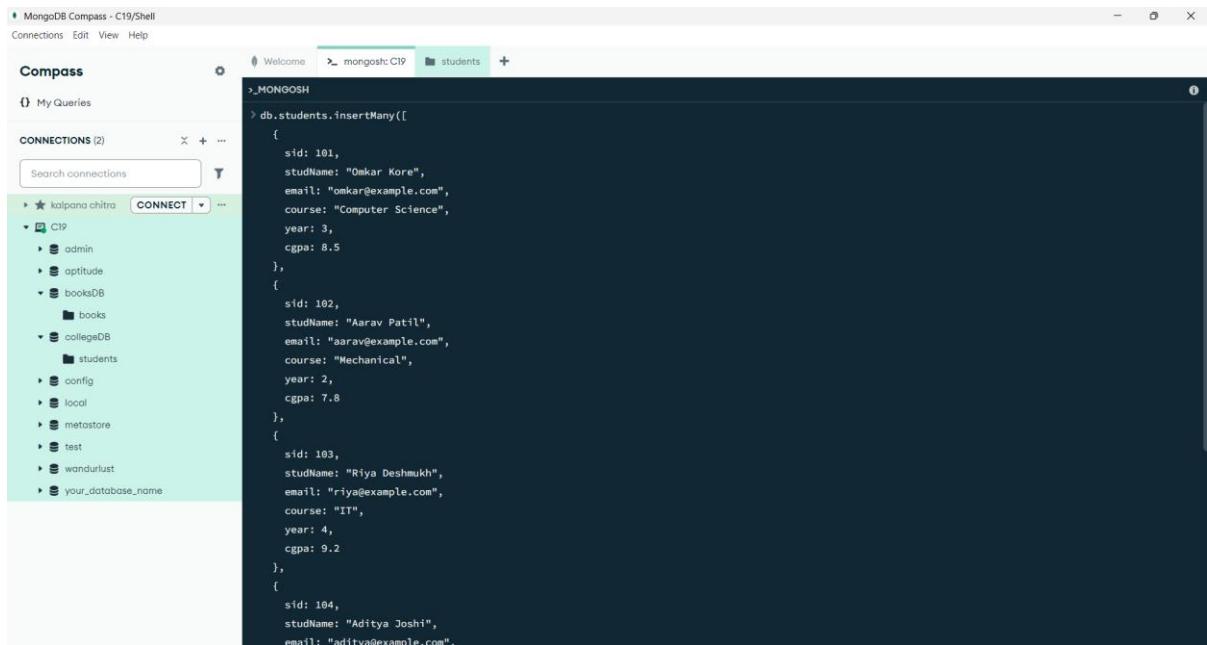
The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists 'C19' as the selected connection. The main area displays the MONGOSH shell with the following command:

```
> db.books.find({ year: { $lt: 1950 } })
< [
  {
    _id: ObjectId('690c71381937ee64df48e641'),
    title: 'The Great Gatsby',
    year: 1925,
    author: 'F. Scott Fitzgerald',
    type: 'Fiction'
  },
  {
    _id: ObjectId('690c71381937ee64df48e645'),
    title: '1984',
    year: 1949,
    author: 'George Orwell',
    type: 'Fiction'
  },
  {
    _id: ObjectId('690c71381937ee64df48e648'),
    title: 'Brave New World',
    year: 1932,
    author: 'Aldous Huxley',
    type: 'Fiction'
  }
]
test>|
```

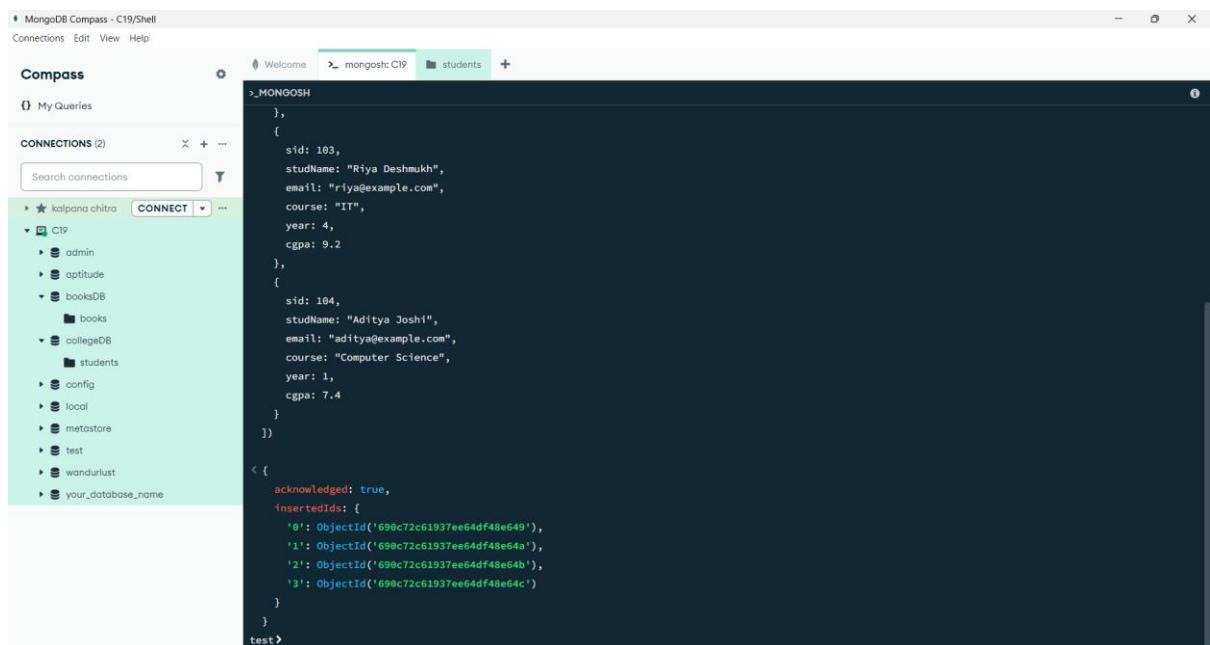
```
db.books.find({ year: { $lt: 1950 } })
```

PART 2:- Student Database

1. INSERTING STUDENTS



```
> MONGOSH
> db.students.insertMany([
  {
    sid: 101,
    studName: "Omkar Kore",
    email: "omkar@example.com",
    course: "Computer Science",
    year: 3,
    cgpa: 8.5
  },
  {
    sid: 102,
    studName: "Aarav Patil",
    email: "aarav@example.com",
    course: "Mechanical",
    year: 2,
    cgpa: 7.8
  },
  {
    sid: 103,
    studName: "Riya Deshmukh",
    email: "riya@example.com",
    course: "IT",
    year: 4,
    cgpa: 9.2
  },
  {
    sid: 104,
    studName: "Aditya Joshi",
    email: "aditya@example.com",
    course: "Computer Science",
    year: 1,
    cgpa: 7.4
  }
])
```



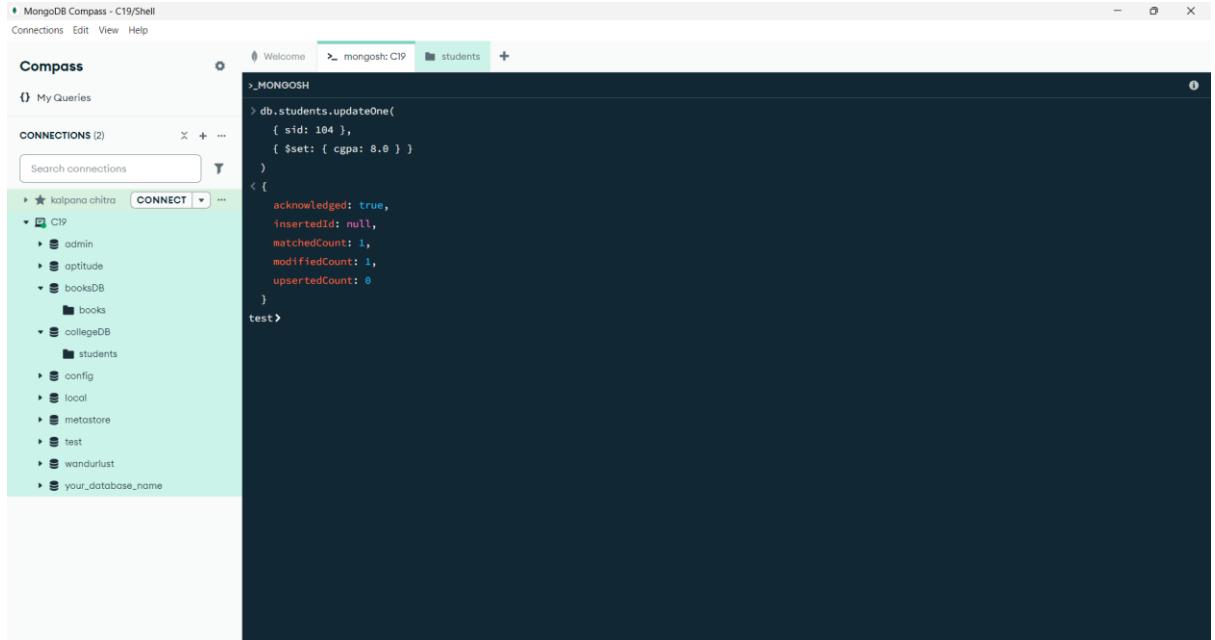
```
> MONGOSH
> db.students.insertMany([
  {
    sid: 101,
    studName: "Omkar Kore",
    email: "omkar@example.com",
    course: "Computer Science",
    year: 3,
    cgpa: 8.5
  },
  {
    sid: 102,
    studName: "Aarav Patil",
    email: "aarav@example.com",
    course: "Mechanical",
    year: 2,
    cgpa: 7.8
  },
  {
    sid: 103,
    studName: "Riya Deshmukh",
    email: "riya@example.com",
    course: "IT",
    year: 4,
    cgpa: 9.2
  },
  {
    sid: 104,
    studName: "Aditya Joshi",
    email: "aditya@example.com",
    course: "Computer Science",
    year: 1,
    cgpa: 7.4
  }
])
< {
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('690c72c61937ee64df48e649'),
    '1': ObjectId('690c72c61937ee64df48e64a'),
    '2': ObjectId('690c72c61937ee64df48e64b'),
    '3': ObjectId('690c72c61937ee64df48e64c')
  ]
}
test>
```

```
db.students.insertMany([
  {
    sid: 101,
    studName: "Omkar Kore",
    email: "omkar@example.com",
    course: "Computer Science",
    year: 3,
    cgpa: 8.5
  },
  {
    sid: 102,
    studName: "Aarav Patil",
    email: "aarav@example.com",
    course: "Mechanical",
    year: 2,
    cgpa: 7.8
  },
  {
    sid: 103,
    studName: "Riya Deshmukh",
    email: "riya@example.com",
    course: "IT",
    year: 4,
    cgpa: 9.2
  },
  {
    sid: 104,
    studName: "Aditya Joshi",
    email: "aditya@example.com",
    course: "Computer Science",
    year: 1,
    cgpa: 7.4
  }
])
```

```
}
```

```
)
```

2. Updating student cgpa



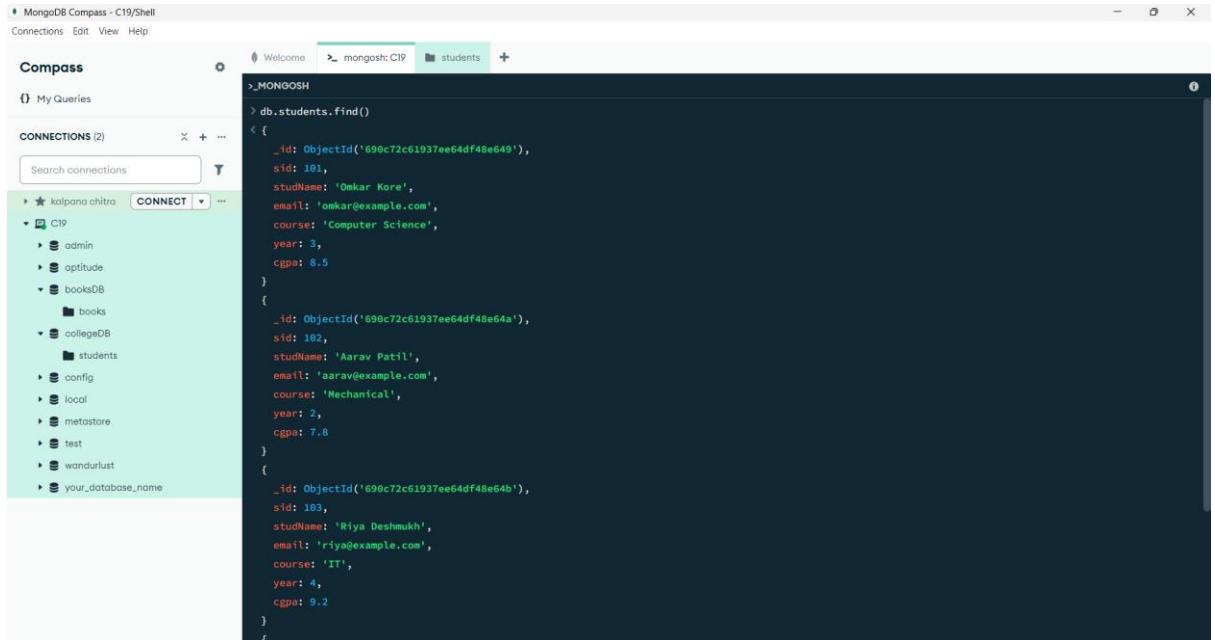
The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists two connections: 'C19' and 'your_database_name'. The 'C19' connection is selected, showing its databases: admin, aptitude, booksDB, collegeDB, config, local, metastore, test, wandurlust, and your_database_name. The main panel displays the MongoDB shell command:

```
> db.students.updateOne(
  { sid: 104 },
  { $set: { cgpa: 8.0 } }
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
test>
```

```
db.students.updateOne(
```

```
  { sid: 104 },
  { $set: { cgpa: 8.0 } }
)
```

3. Get all students

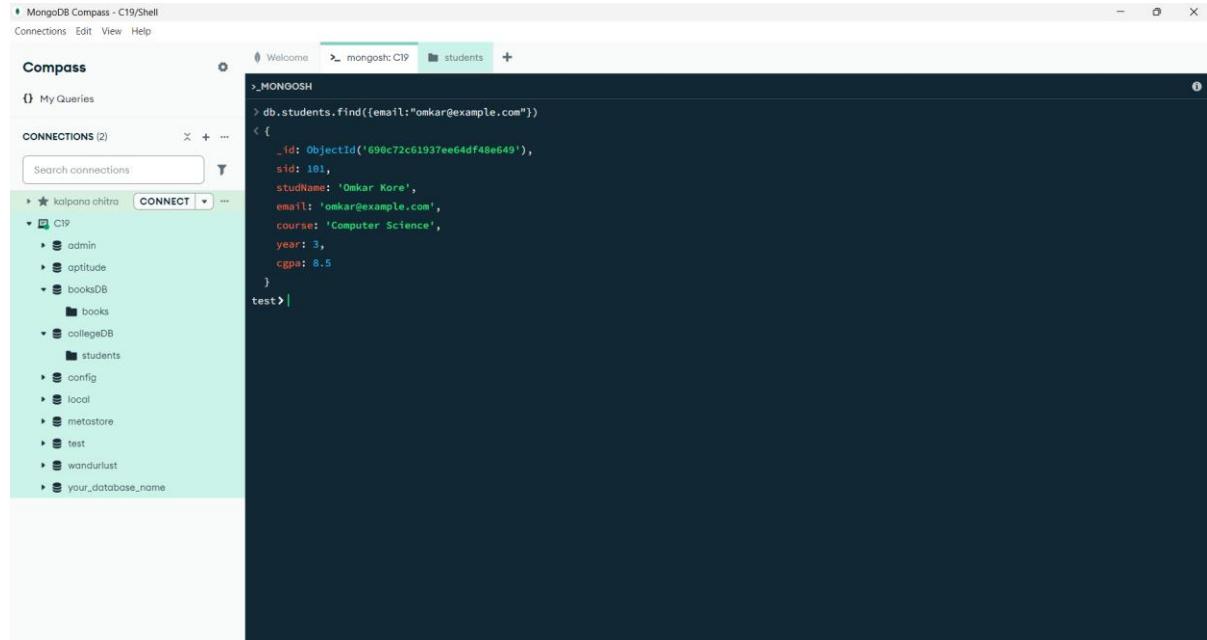


The screenshot shows the MongoDB Compass interface with the 'C19' connection selected. The main panel displays the MongoDB shell command:

```
> db.students.find()
< [
  {
    _id: ObjectId('690c72c61937ee64df48e649'),
    sid: 101,
    studName: 'Omkar Kore',
    email: 'omkar@example.com',
    course: 'Computer Science',
    year: 3,
    cgpa: 8.5
  },
  {
    _id: ObjectId('690c72c61937ee64df48e64a'),
    sid: 102,
    studName: 'Aarav Patil',
    email: 'aarav@example.com',
    course: 'Mechanical',
    year: 2,
    cgpa: 7.8
  },
  {
    _id: ObjectId('690c72c61937ee64df48e64b'),
    sid: 103,
    studName: 'Riya Deshmukh',
    email: 'riya@example.com',
    course: 'IT',
    year: 4,
    cgpa: 9.2
  }
]
```

```
db.students.find()
```

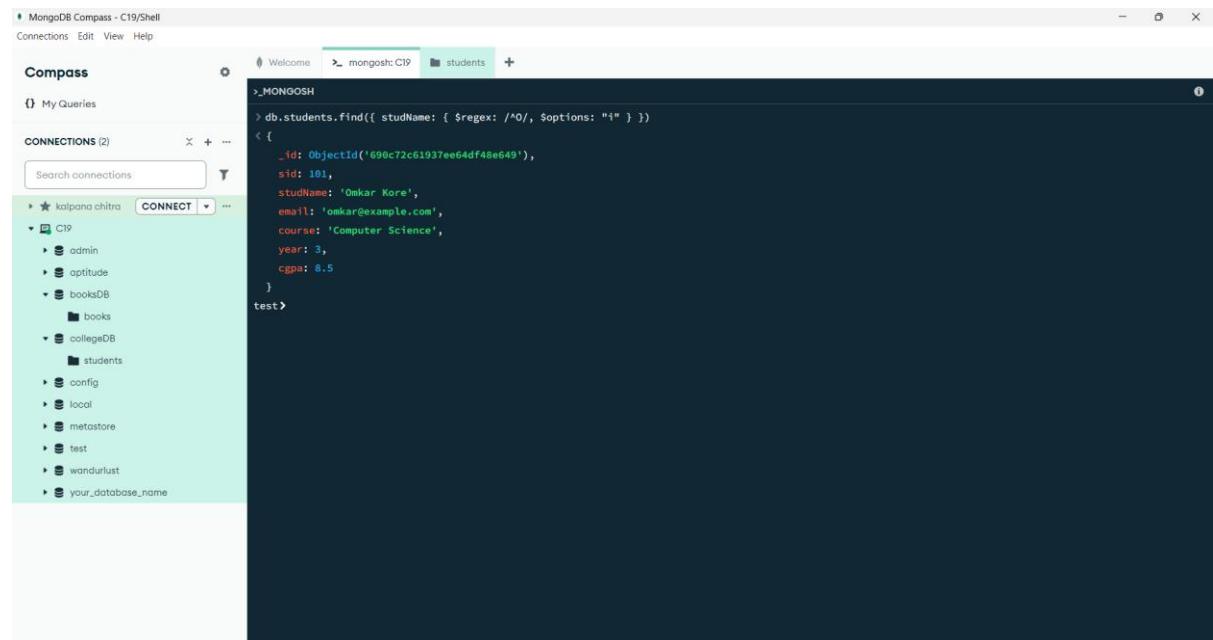
4. Get student with email omkar@example.com



```
>_MONGOSH
> db.students.find({email:"omkar@example.com"})
< [
  {
    _id: ObjectId('690c72c61937ee64df48e649'),
    sid: 101,
    studName: 'Omkar Kore',
    email: 'omkar@example.com',
    course: 'Computer Science',
    year: 3,
    cgpa: 8.5
  }
]
test>
```

```
db.students.find({email:"omkar@example.com"})
```

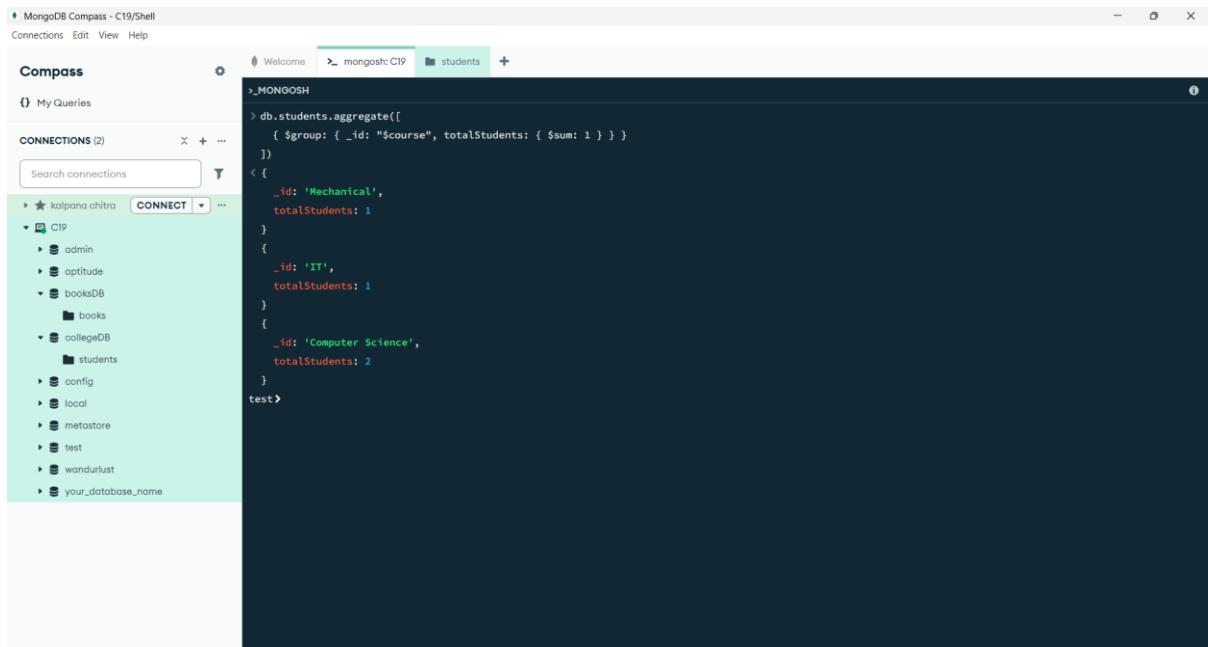
5. Find names starting with O



```
>_MONGOSH
> db.students.find({ studName: { $regex: /^O/, $options: "i" } })
< [
  {
    _id: ObjectId('690c72c61937ee64df48e649'),
    sid: 101,
    studName: 'Omkar Kore',
    email: 'omkar@example.com',
    course: 'Computer Science',
    year: 3,
    cgpa: 8.5
  }
]
test>
```

```
db.students.find({ studName: { $regex: /^O/, $options: "i" } })
```

6. Count students in each course

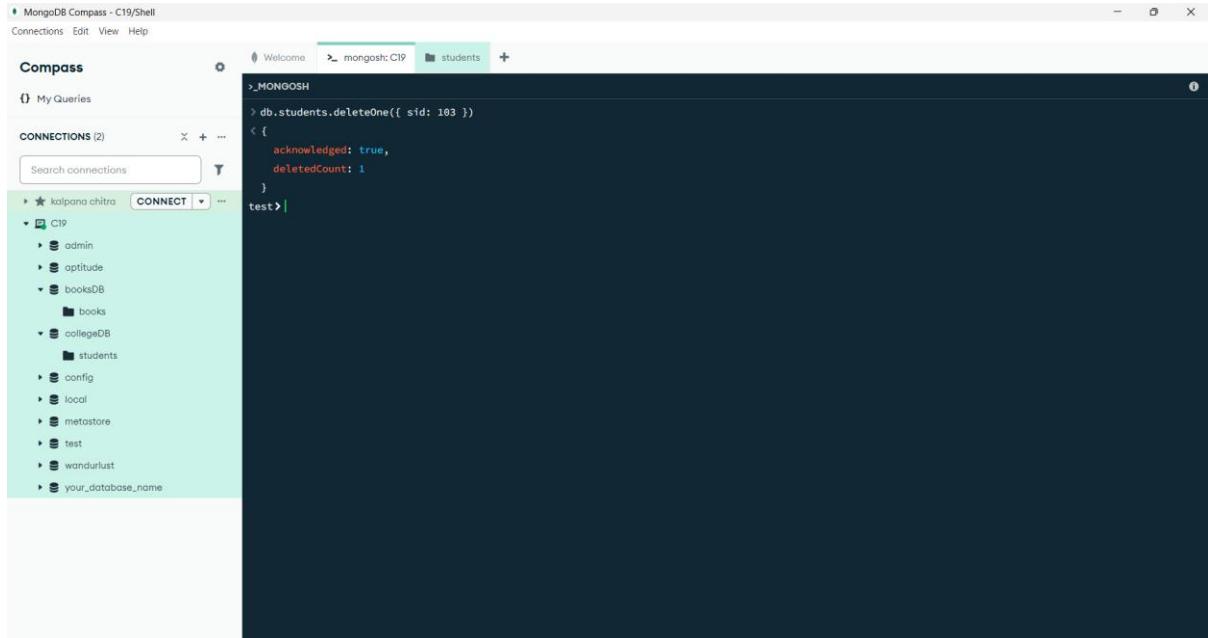


The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists 'C19' as the selected connection, which contains databases like 'admin', 'aptitude', 'booksDB', 'collegeDB', 'config', 'local', 'metastore', 'test', and 'wandurlust'. The main panel displays the MongoDB shell command:

```
>_MONGOSH
>db.students.aggregate([
  { $group: { _id: "$course", totalStudents: { $sum: 1 } } }
])
< [
  {
    _id: 'Mechanical',
    totalStudents: 1
  },
  {
    _id: 'IT',
    totalStudents: 1
  },
  {
    _id: 'Computer Science',
    totalStudents: 2
  }
]
test>
```

```
db.students.aggregate([
  { $group: { _id: "$course", totalStudents: { $sum: 1 } } }
])
```

7. Delete a student

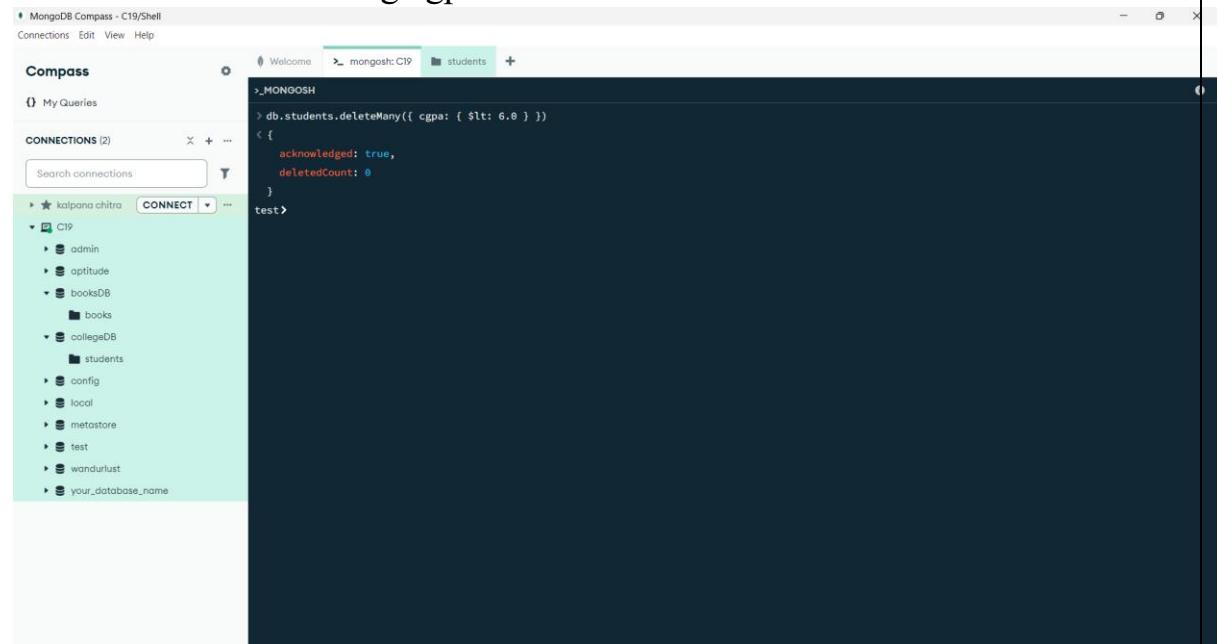


The screenshot shows the MongoDB Compass interface with 'C19' selected. The main panel displays the MongoDB shell command:

```
>_MONGOSH
>db.students.deleteOne({ sid: 103 })
< {
  acknowledged: true,
  deletedCount: 1
}
test>
```

```
db.students.deleteOne({ sid: 102 })
```

8. Delete all students having cgpa below 6



The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists two connections: 'kalpana chitra' and 'C19'. The 'C19' connection is selected, and its database list includes 'admin', 'aptitude', 'booksDB' (which contains 'books'), 'collegeDB' (which contains 'students'), 'config', 'local', 'metastore', 'test', 'wandurlust', and 'your_database_name'. The main panel displays a mongo shell command and its response:

```
> db.students.deleteMany({ cgpa: { $lt: 6.0 } })  
< {  
  acknowledged: true,  
  deletedCount: 0  
}  
test>
```

```
db.students.deleteMany({ cgpa: { $lt: 6.0 } })
```