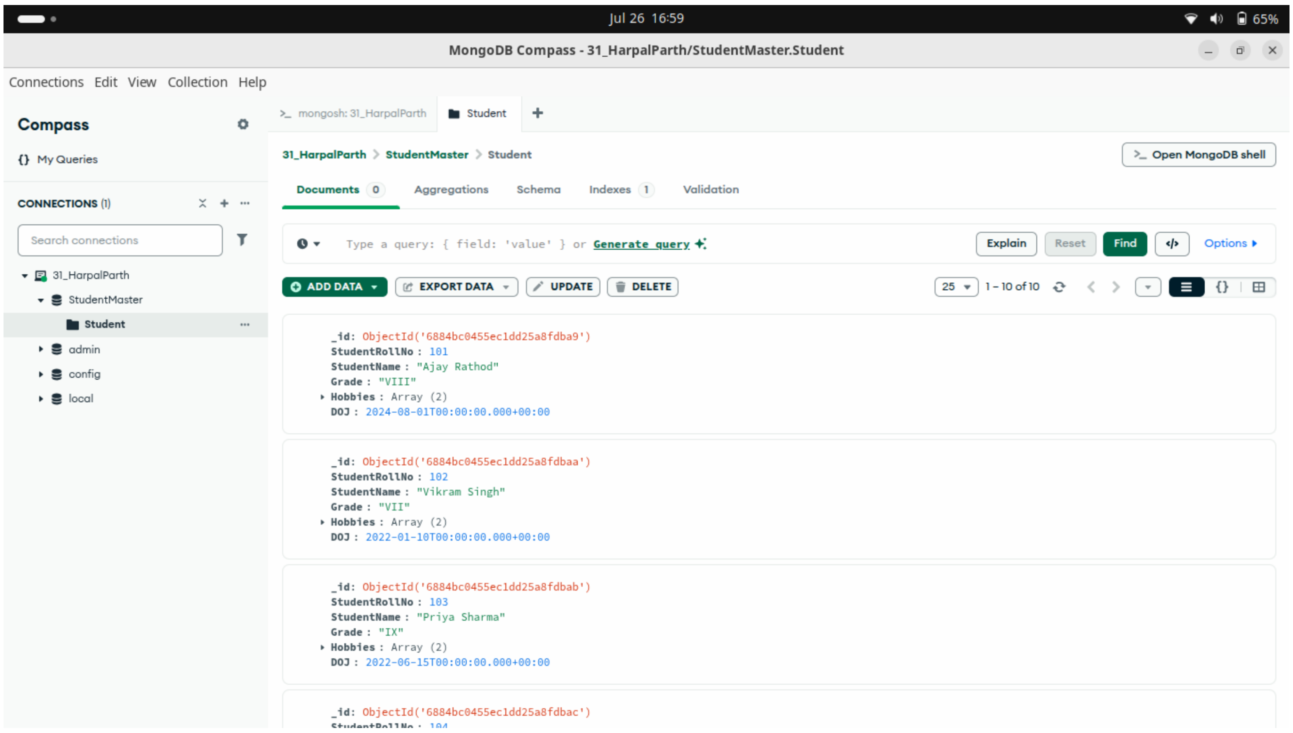
Name : Parth Harpal Roll No: 24CI2110031

BDT Practical Assignment – 1

Q-1) Create a StudentMaster database with a collection called “Student” containing documents with some or all of the following fields: StudentRollNo, StudentName, Grade, Hobbies, and DOJ.

Perform the following operations on the database:

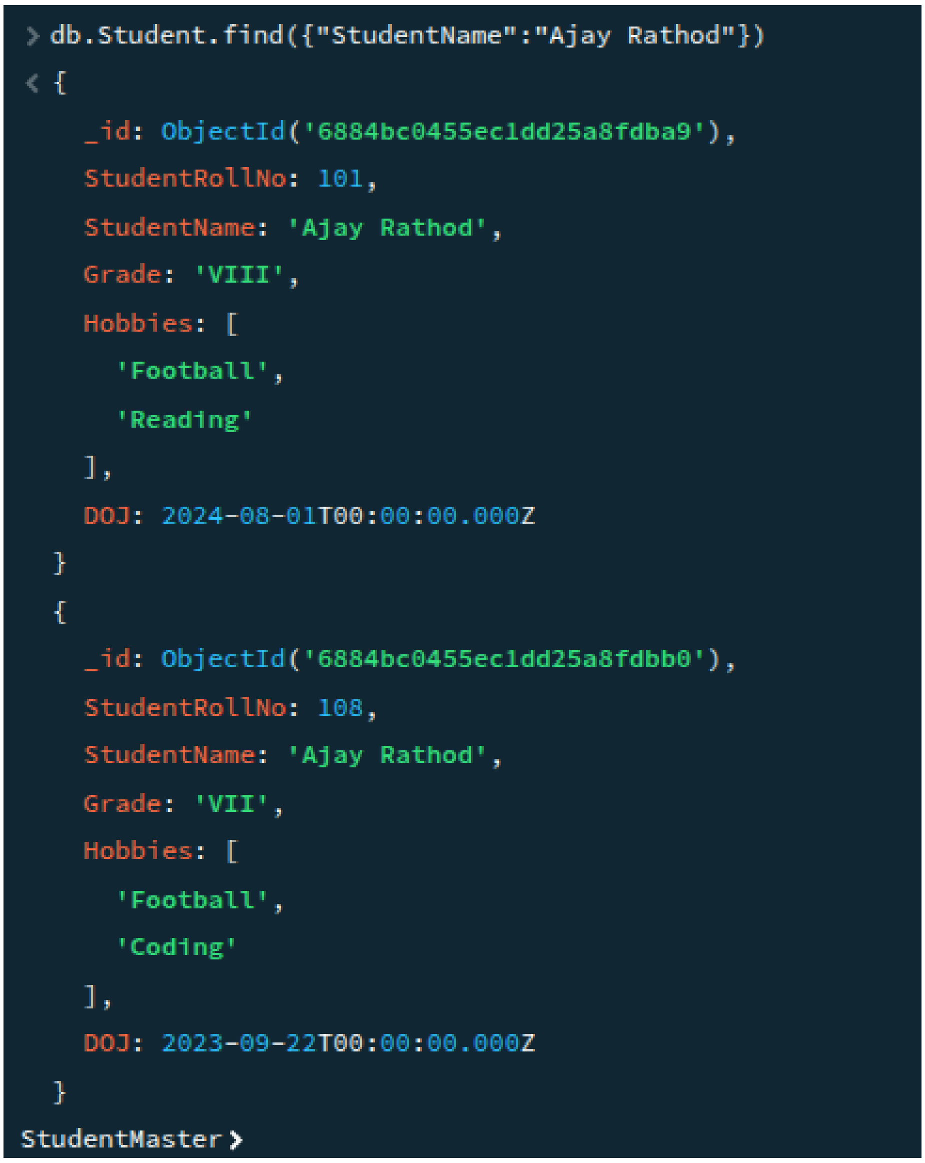
1. Insert 10 Records in the database.



```
_id: ObjectId('6884bc0455ec1dd25a8fdba9')
StudentRollNo : 101
StudentName : "Ajay Rathod"
Grade : "VIII"
Hobbies : Array (2)
DOJ : 2024-08-01T00:00:00.000+00:00

_id: ObjectId('6884bc0455ec1dd25a8fdbaa')
StudentRollNo : 102
StudentName : "Vikram Singh"
Grade : "VII"
Hobbies : Array (2)
DOJ : 2022-01-10T00:00:00.000+00:00

_id: ObjectId('6884bc0455ec1dd25a8fdbab')
StudentRollNo : 103
StudentName : "Priya Sharma"
Grade : "IX"
Hobbies : Array (2)
DOJ : 2022-06-15T00:00:00.000+00:00

_id: ObjectId('6884bc0455ec1dd25a8fdbac')
StudentRollNo : 104
```

Name : Parth Harpal　　　　　　　　　　　　　　　　Roll No: 24CI2110031

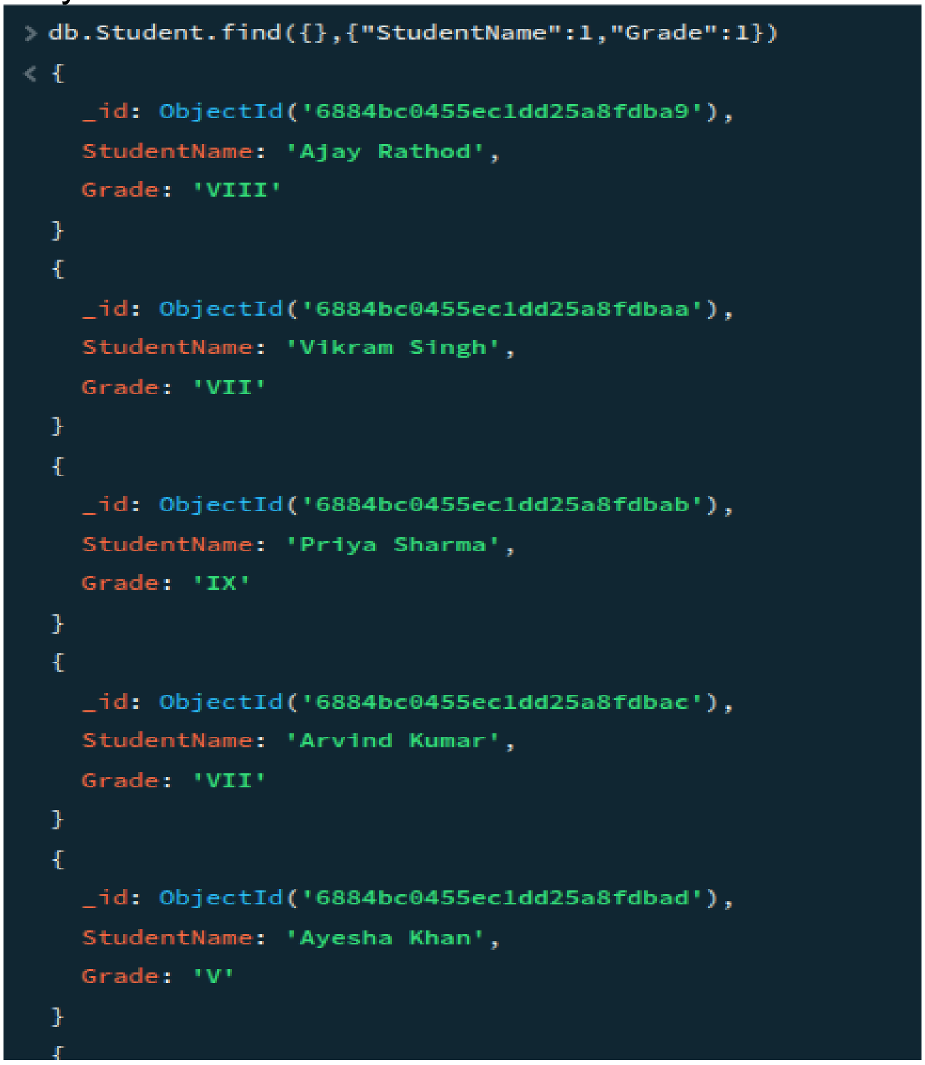
2. Find the document wherein the “StudName” has value “Ajay Rathod”.



```
> db.Student.find({"StudentName":"Ajay Rathod"})
< {
    _id: ObjectId('6884bc0455ec1dd25a8fdba9'),
    StudentRollNo: 101,
    StudentName: 'Ajay Rathod',
    Grade: 'VIII',
    Hobbies: [
      'Football',
      'Reading'
    ],
    DOJ: 2024-08-01T00:00:00.000Z
  }
  {
    _id: ObjectId('6884bc0455ec1dd25a8fdbb0'),
    StudentRollNo: 108,
    StudentName: 'Ajay Rathod',
    Grade: 'VII',
    Hobbies: [
      'Football',
      'Coding'
    ],
    DOJ: 2023-09-22T00:00:00.000Z
  }
StudentMaster>
```

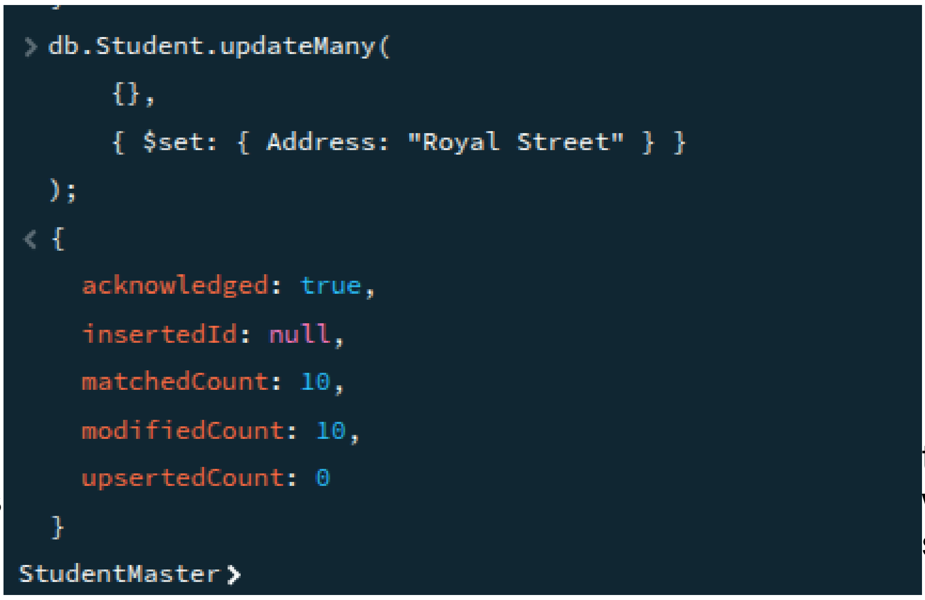
Name : Parth Harpal Roll No: 24CI2110031

3. Retrieve only Student Name and Grade.



```
> db.Student.find({},{"StudentName":1,"Grade":1})
< {
    _id: ObjectId('6884bc0455ec1dd25a8fdba9'),
    StudentName: 'Ajay Rathod',
    Grade: 'VIII'
  }
  {
    _id: ObjectId('6884bc0455ec1dd25a8fdbaa'),
    StudentName: 'Vikram Singh',
    Grade: 'VII'
  }
  {
    _id: ObjectId('6884bc0455ec1dd25a8fdbab'),
    StudentName: 'Priya Sharma',
    Grade: 'IX'
  }
  {
    _id: ObjectId('6884bc0455ec1dd25a8fdbac'),
    StudentName: 'Arvind Kumar',
    Grade: 'VII'
  }
  {
    _id: ObjectId('6884bc0455ec1dd25a8fdbad'),
    StudentName: 'Ayesha Khan',
    Grade: 'V'
  }
  {
```

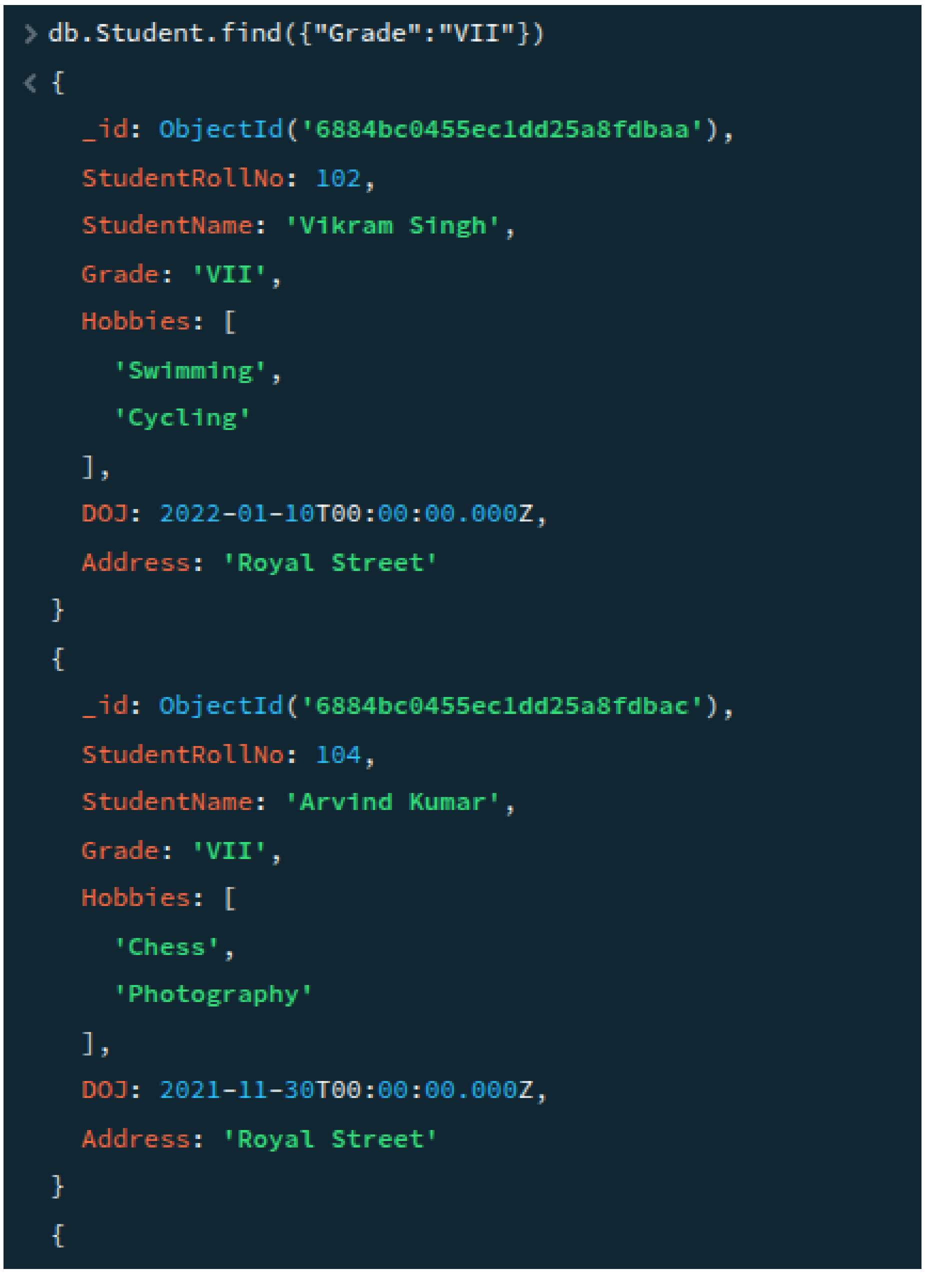
4. Add new field “Address” in Student Collection.



```
> db.Student.updateMany(
      {},
      { $set: { Address: "Royal Street" } }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 10,
    modifiedCount: 10,
    upsertedCount: 0
  }
StudentMaster>
```

5. Find those documents where the Grade is set to ‘VII’.

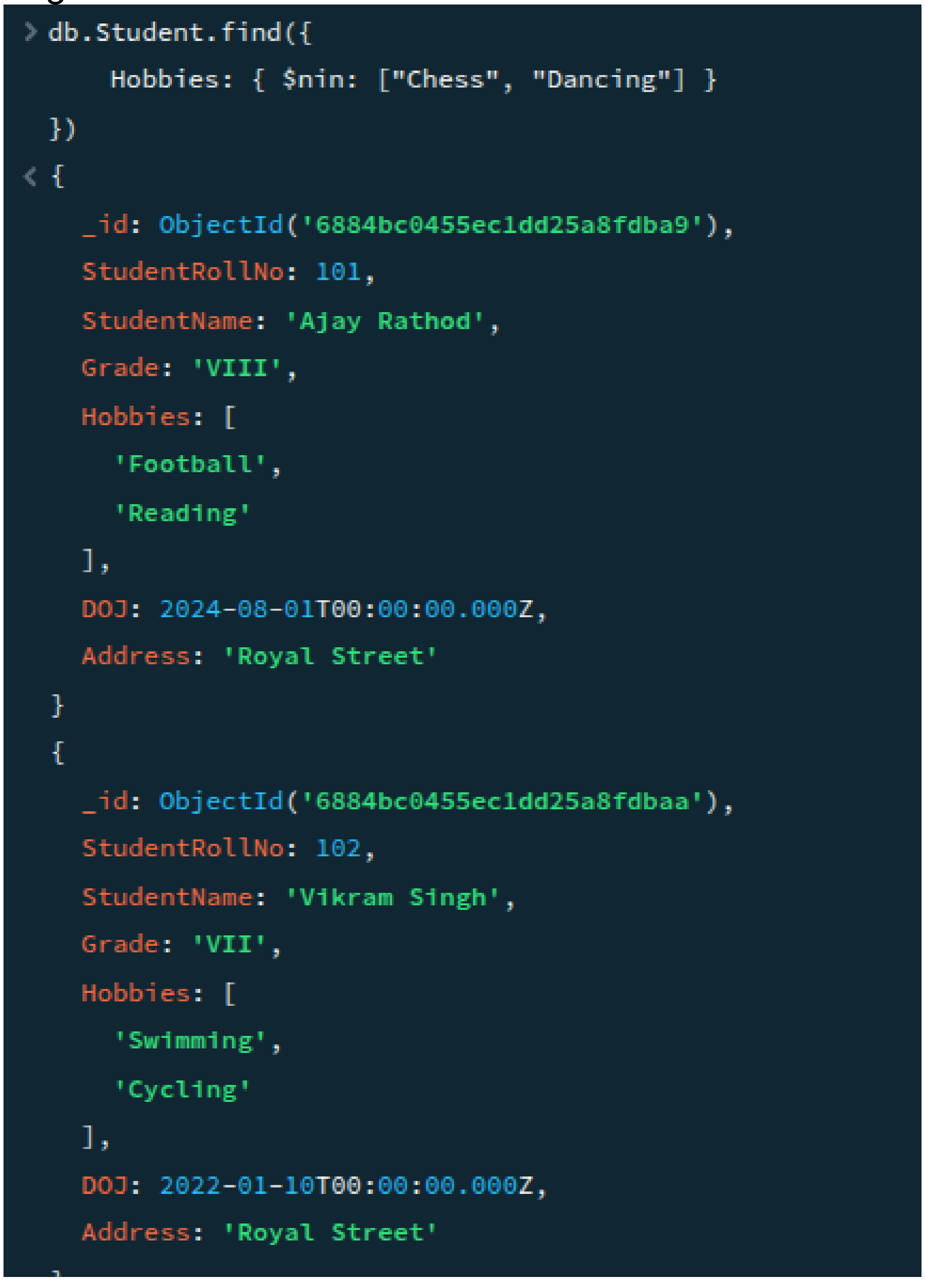
Name : Parth Harpal Roll No: 24CI2110031



```
> db.Student.find({"Grade":"VII"})
< {
    _id: ObjectId('6884bc0455ec1dd25a8fdbaa'),
    StudentRollNo: 102,
    StudentName: 'Vikram Singh',
    Grade: 'VII',
    Hobbies: [
      'Swimming',
      'Cycling'
    ],
    DOJ: 2022-01-10T00:00:00.000Z,
    Address: 'Royal Street'
  }
  {
    _id: ObjectId('6884bc0455ec1dd25a8fdbac'),
    StudentRollNo: 104,
    StudentName: 'Arvind Kumar',
    Grade: 'VII',
    Hobbies: [
      'Chess',
      'Photography'
    ],
    DOJ: 2021-11-30T00:00:00.000Z,
    Address: 'Royal Street'
  }
  {
```

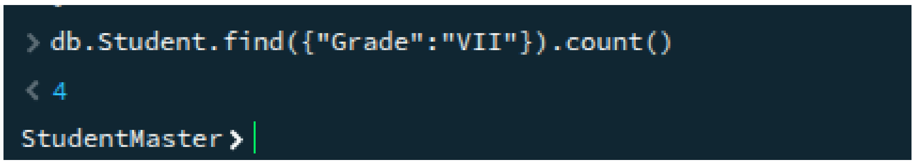
Name : Parth Harpal   Roll No: 24CI2110031

6. Find those documents where the Hobbies is set neither to ‘Chess’ nor is set to ‘Dancing”.



```
> db.Student.find({
      Hobbies: { $nin: ["Chess", "Dancing"] }
  })
< {
    _id: ObjectId('6884bc0455ec1dd25a8fdba9'),
    StudentRollNo: 101,
    StudentName: 'Ajay Rathod',
    Grade: 'VIII',
    Hobbies: [
      'Football',
      'Reading'
    ],
    DOJ: 2024-08-01T00:00:00.000Z,
    Address: 'Royal Street'
  }
  {
    _id: ObjectId('6884bc0455ec1dd25a8fdbaa'),
    StudentRollNo: 102,
    StudentName: 'Vikram Singh',
    Grade: 'VII',
    Hobbies: [
      'Swimming',
      'Cycling'
    ],
    DOJ: 2022-01-10T00:00:00.000Z,
    Address: 'Royal Street'
```

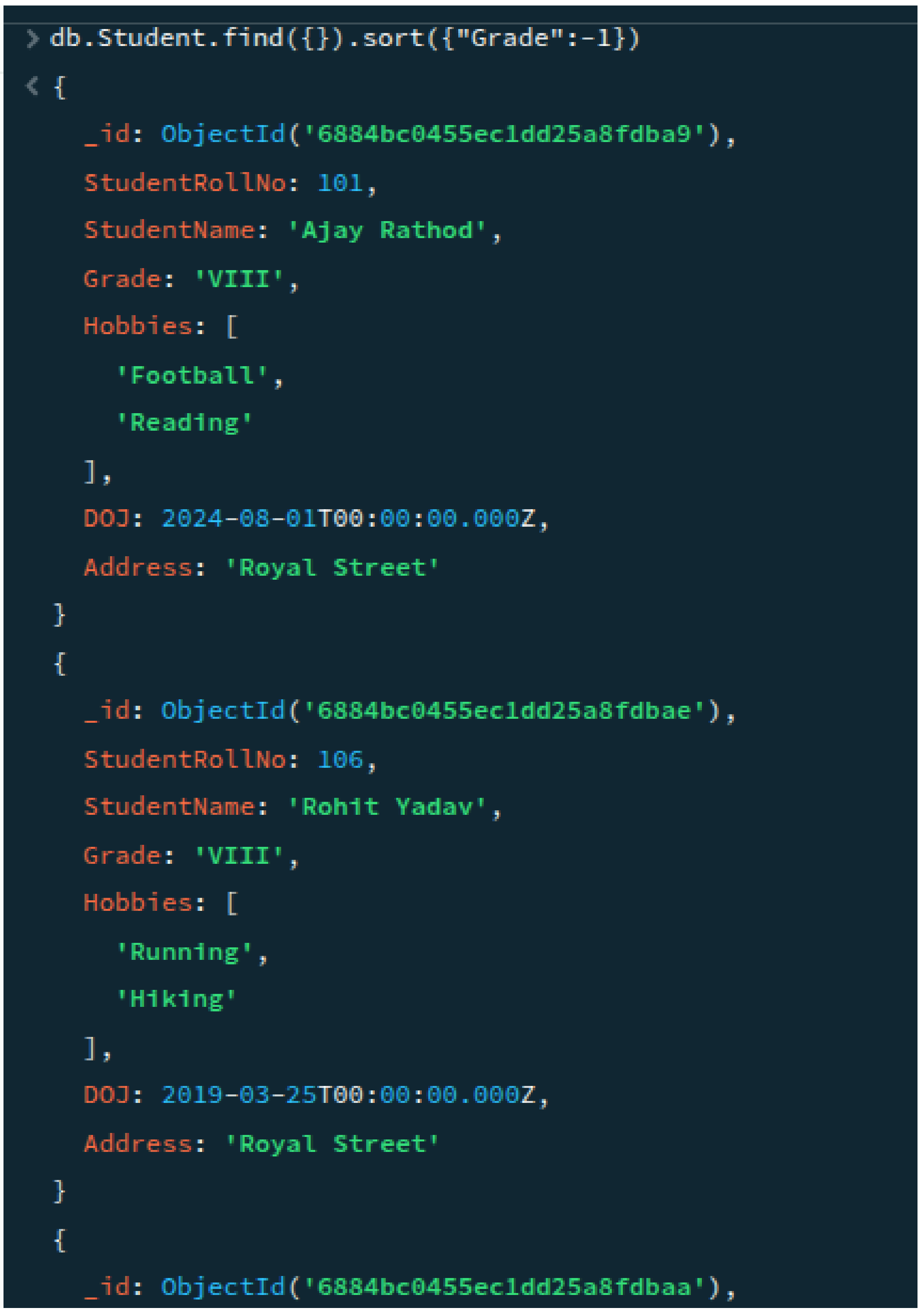
7. Find total the number of documents where Grade is ‘VII’.



```
> db.Student.find({"Grade":"VII"}).count()
< 4
StudentMaster>
```

8. Sort the documents in descending order of Grade.

Name : Parth Harpal Roll No: 24CI2110031



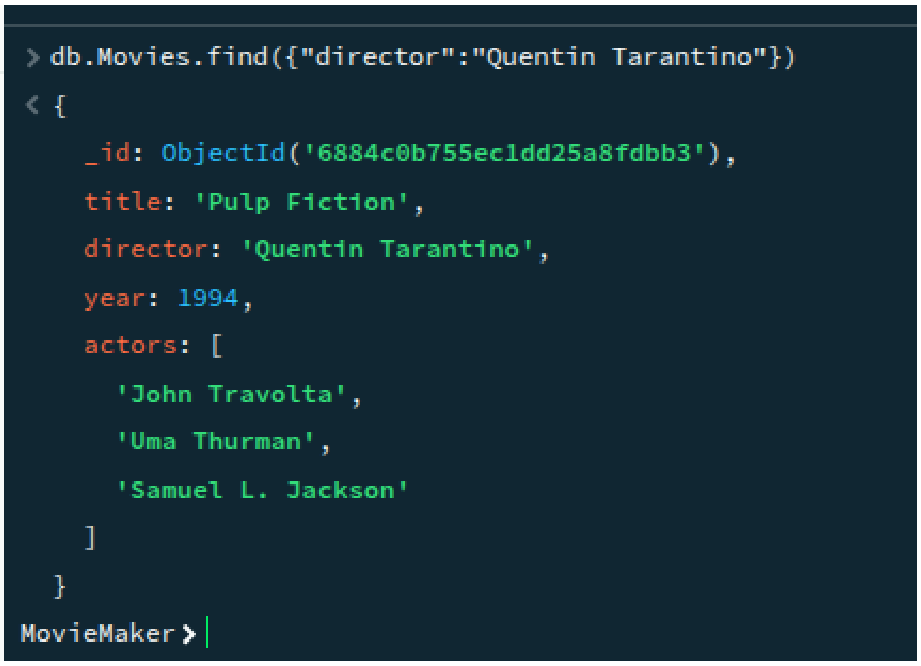
```
> db.Student.find({}).sort({"Grade":-1})
< {
    _id: ObjectId('6884bc0455ec1dd25a8fdba9'),
    StudentRollNo: 101,
    StudentName: 'Ajay Rathod',
    Grade: 'VIII',
    Hobbies: [
      'Football',
      'Reading'
    ],
    DOJ: 2024-08-01T00:00:00.000Z,
    Address: 'Royal Street'
  }
  {
    _id: ObjectId('6884bc0455ec1dd25a8fdbae'),
    StudentRollNo: 106,
    StudentName: 'Rohit Yadav',
    Grade: 'VIII',
    Hobbies: [
      'Running',
      'Hiking'
    ],
    DOJ: 2019-03-25T00:00:00.000Z,
    Address: 'Royal Street'
  }
  {
    _id: ObjectId('6884bc0455ec1dd25a8fdbaa'),
```

Name : Parth Harpal Roll No: 24CI2110031

Q-2) Create a MovieMaker Database with a collection called “Movies” containing documents with some or all the following fields: titles, directors, years, actors.

Perform the following operations on the database:

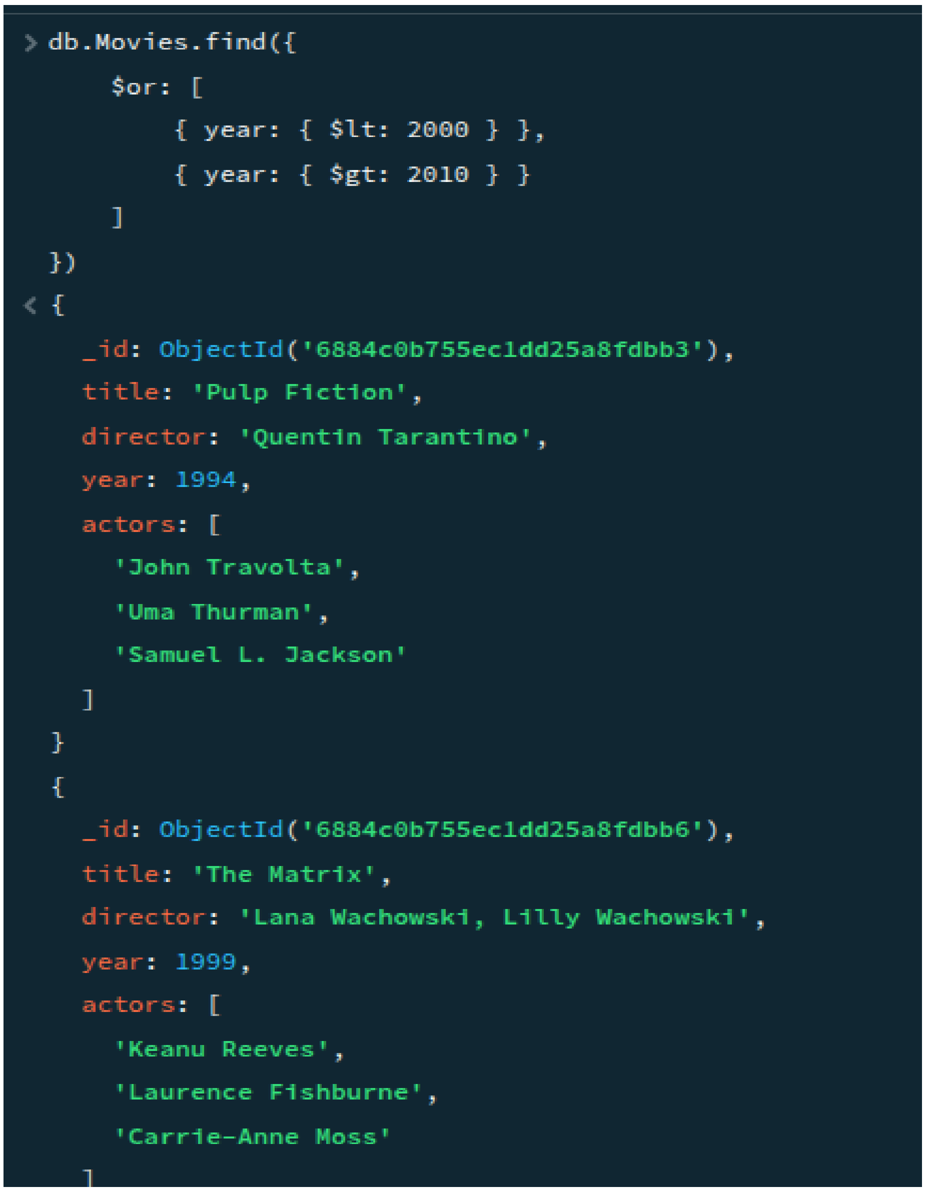
1. Retrieve all documents with Director set to &quot;Quentin Tarantino&quot;



```
> db.Movies.find({"director":"Quentin Tarantino"})
< {
    _id: ObjectId('6884c0b755ec1dd25a8fdbb3'),
    title: 'Pulp Fiction',
    director: 'Quentin Tarantino',
    year: 1994,
    actors: [
      'John Travolta',
      'Uma Thurman',
      'Samuel L. Jackson'
    ]
  }
MovieMaker>
```

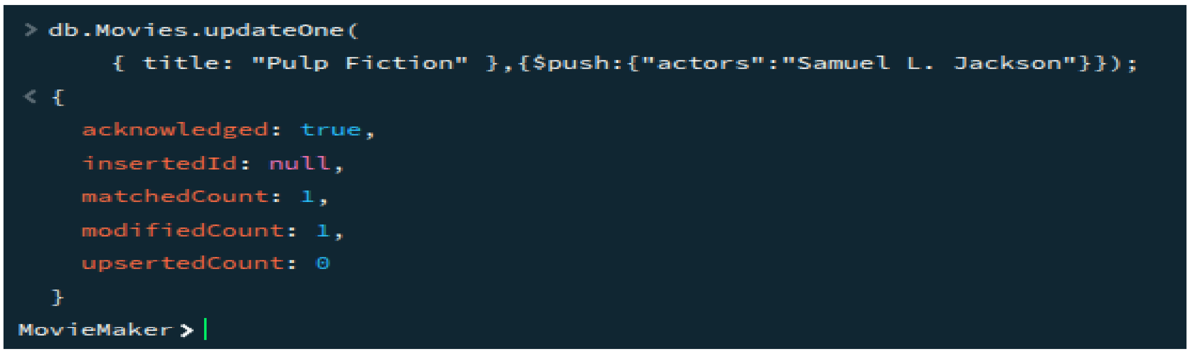
Name : Parth Harpal Roll No: 24CI2110031

2. Retrieve all movies released before the year 2000 or after 2010



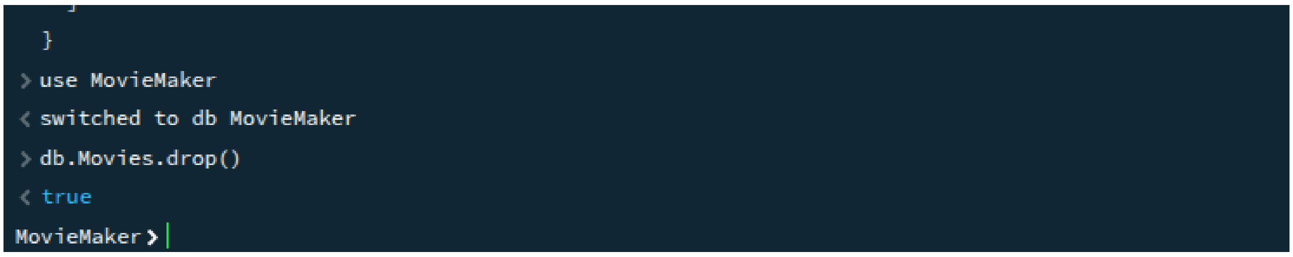
```
> db.Movies.find({
      $or: [
          { year: { $lt: 2000 } },
          { year: { $gt: 2010 } }
      ]
  })
< {
    _id: ObjectId('6884c0b755ec1dd25a8fdbb3'),
    title: 'Pulp Fiction',
    director: 'Quentin Tarantino',
    year: 1994,
    actors: [
      'John Travolta',
      'Uma Thurman',
      'Samuel L. Jackson'
    ]
  }
  {
    _id: ObjectId('6884c0b755ec1dd25a8fdbb6'),
    title: 'The Matrix',
    director: 'Lana Wachowski, Lilly Wachowski',
    year: 1999,
    actors: [
      'Keanu Reeves',
      'Laurence Fishburne',
      'Carrie-Anne Moss'
    ]
```

3. Add an actor named Samuel L. Jackson to the movie Pulp Fiction



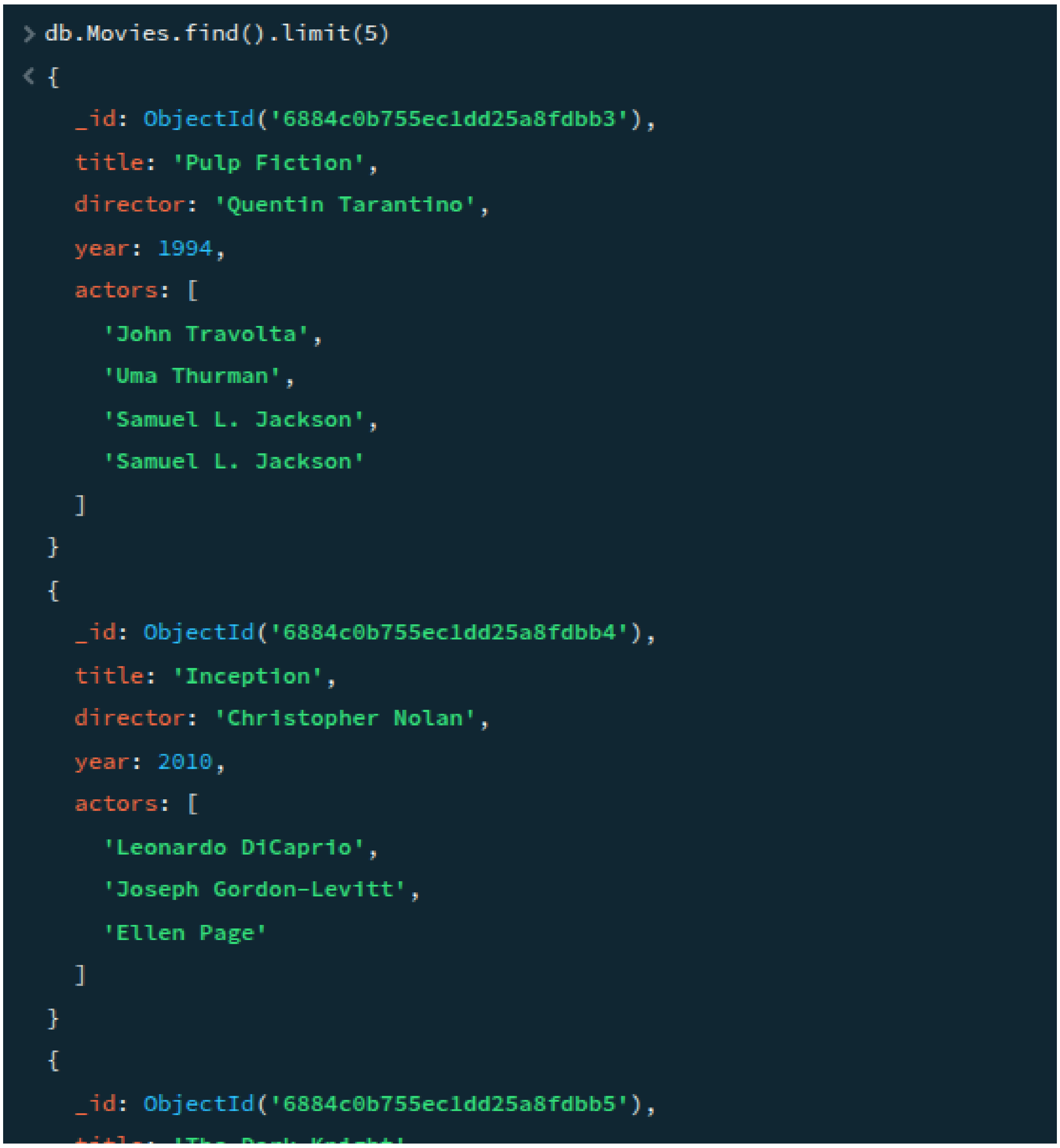
```
> db.Movies.updateOne(
      { title: "Pulp Fiction" },{$push:{"actors":"Samuel L. Jackson"}});
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
MovieMaker>
```

4.Delete all movies.



```
  }
> use MovieMaker
< switched to db MovieMaker
> db.Movies.drop()
< true
MovieMaker >
```

5. Display first 5 movies.

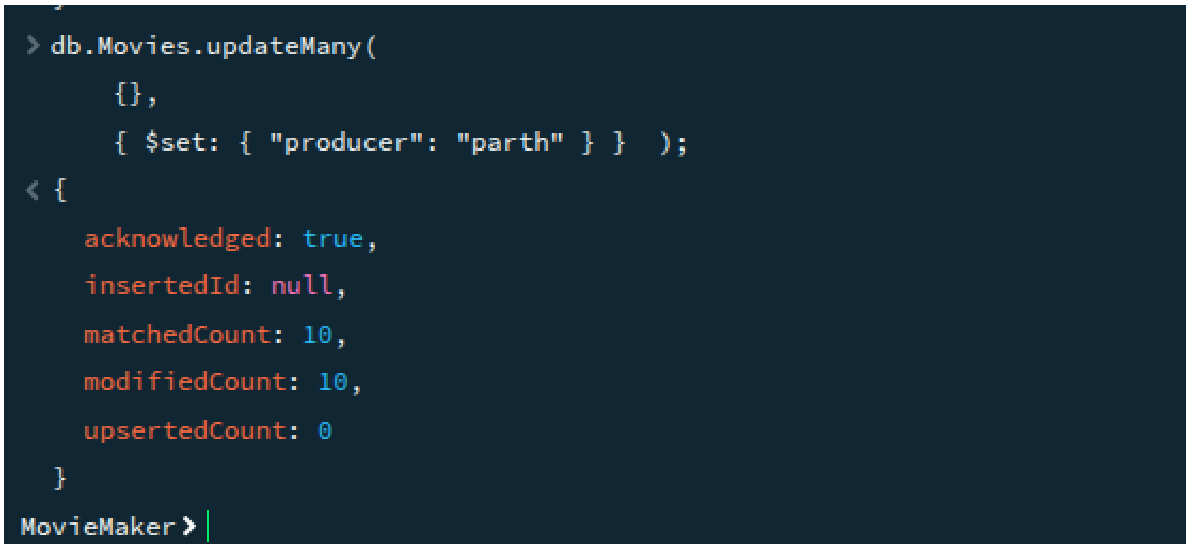


```
> db.Movies.find().limit(5)
< {
    _id: ObjectId('6884c0b755ec1dd25a8fdbb3'),
    title: 'Pulp Fiction',
    director: 'Quentin Tarantino',
    year: 1994,
    actors: [
      'John Travolta',
      'Uma Thurman',
      'Samuel L. Jackson',
      'Samuel L. Jackson'
    ]
  }
  {
    _id: ObjectId('6884c0b755ec1dd25a8fdbb4'),
    title: 'Inception',
    director: 'Christopher Nolan',
    year: 2010,
    actors: [
      'Leonardo DiCaprio',
      'Joseph Gordon-Levitt',
      'Ellen Page'
    ]
  }
  {
    _id: ObjectId('6884c0b755ec1dd25a8fdbb5'),
```

6. Add new field 'Producer' to the collection.

Name : Parth Harpal Roll No: 24CI2110031

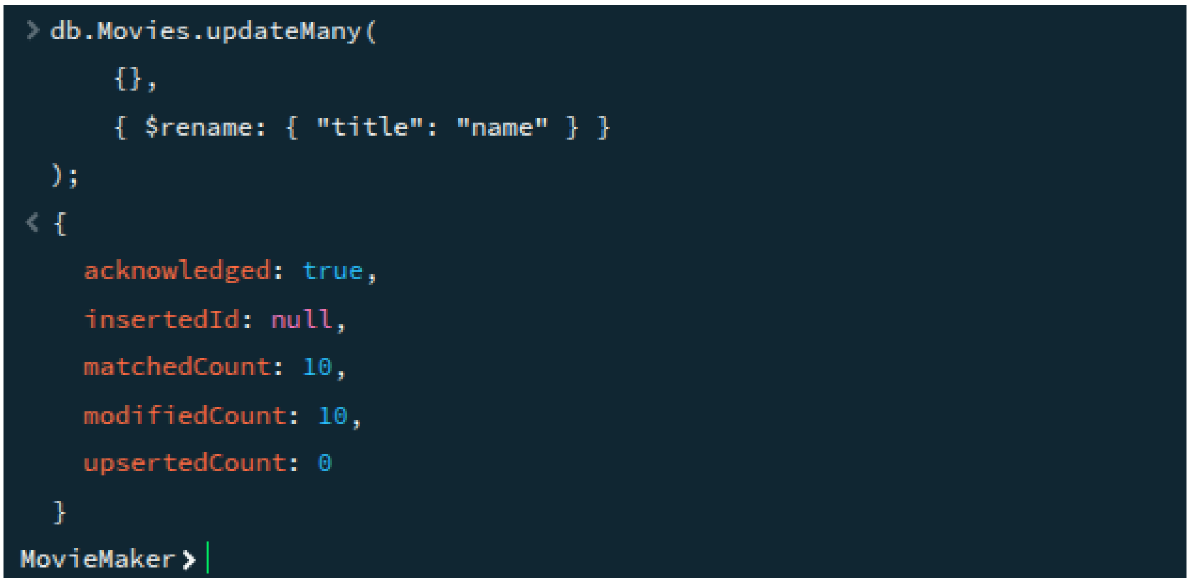
7.



```
> db.Movies.updateMany(
      {},
      { $set: { "producer": "parth" } }  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 10,
    modifiedCount: 10,
    upsertedCount: 0
  }
MovieMaker>
```

Export the collection to movies. json file.

8. Rename the field ‘titles’ to ‘name’.



```
> db.Movies.updateMany(
      {},
      { $rename: { "title": "name" } }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 10,
    modifiedCount: 10,
    upsertedCount: 0
  }
MovieMaker>
```

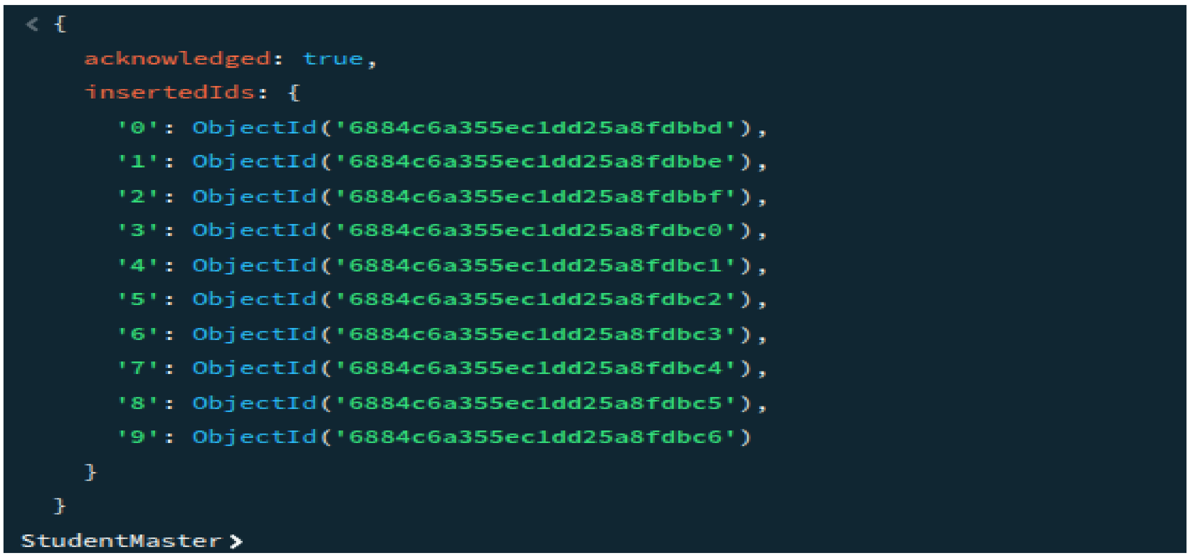
Q=3) Create a StudentMaster database with a collection called “Student” containing documents with some or all of the following fields: StudentRollNo, StudentName, Grade, Hobbies, and DOJ.

Perform the following operations on the database:

1. Insert 10 Records in the database.

Name : Parth Harpal Roll No: 24CI2110031

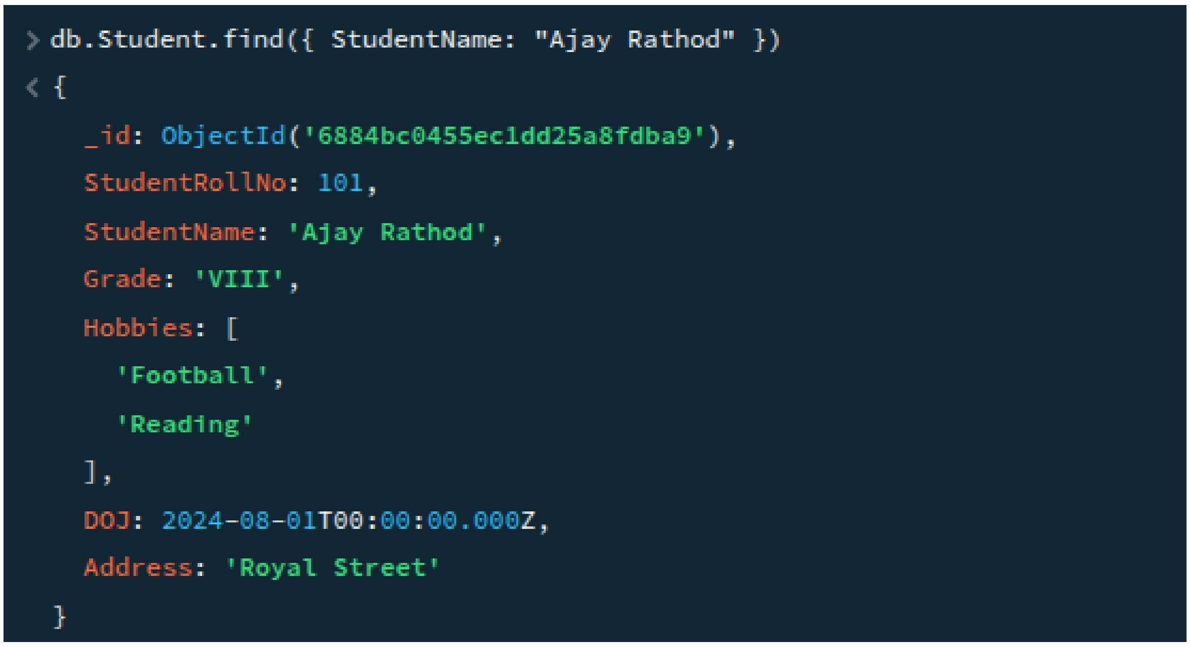
2.



```
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('6884c6a355ec1dd25a8fdbbd'),
      '1': ObjectId('6884c6a355ec1dd25a8fdbbe'),
      '2': ObjectId('6884c6a355ec1dd25a8fdbbf'),
      '3': ObjectId('6884c6a355ec1dd25a8fdbc0'),
      '4': ObjectId('6884c6a355ec1dd25a8fdbc1'),
      '5': ObjectId('6884c6a355ec1dd25a8fdbc2'),
      '6': ObjectId('6884c6a355ec1dd25a8fdbc3'),
      '7': ObjectId('6884c6a355ec1dd25a8fdbc4'),
      '8': ObjectId('6884c6a355ec1dd25a8fdbc5'),
      '9': ObjectId('6884c6a355ec1dd25a8fdbc6')
    }
  }
StudentMaster >
```

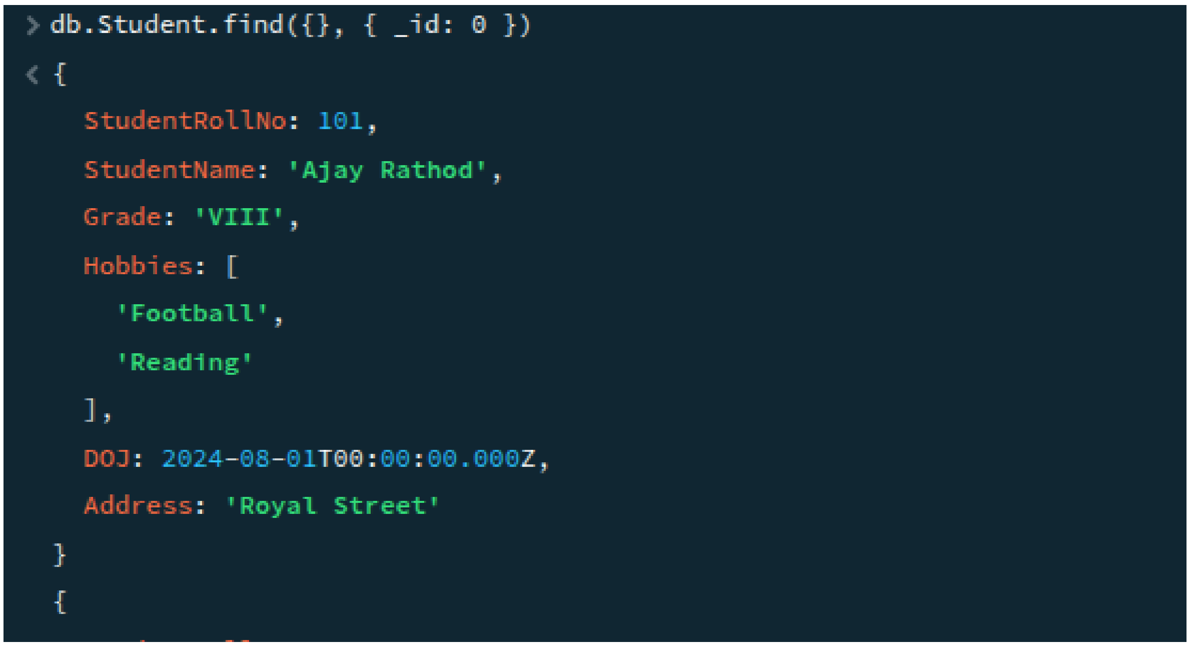
Find the document wherein the “StudName” has value “Ajay Rathod”.

3.



```
> db.Student.find({ StudentName: "Ajay Rathod" })
< {
    _id: ObjectId('6884bc0455ec1dd25a8fdba9'),
    StudentRollNo: 101,
    StudentName: 'Ajay Rathod',
    Grade: 'VIII',
    Hobbies: [
      'Football',
      'Reading'
    ],
    DOJ: 2024-08-01T00:00:00.000Z,
    Address: 'Royal Street'
  }
```

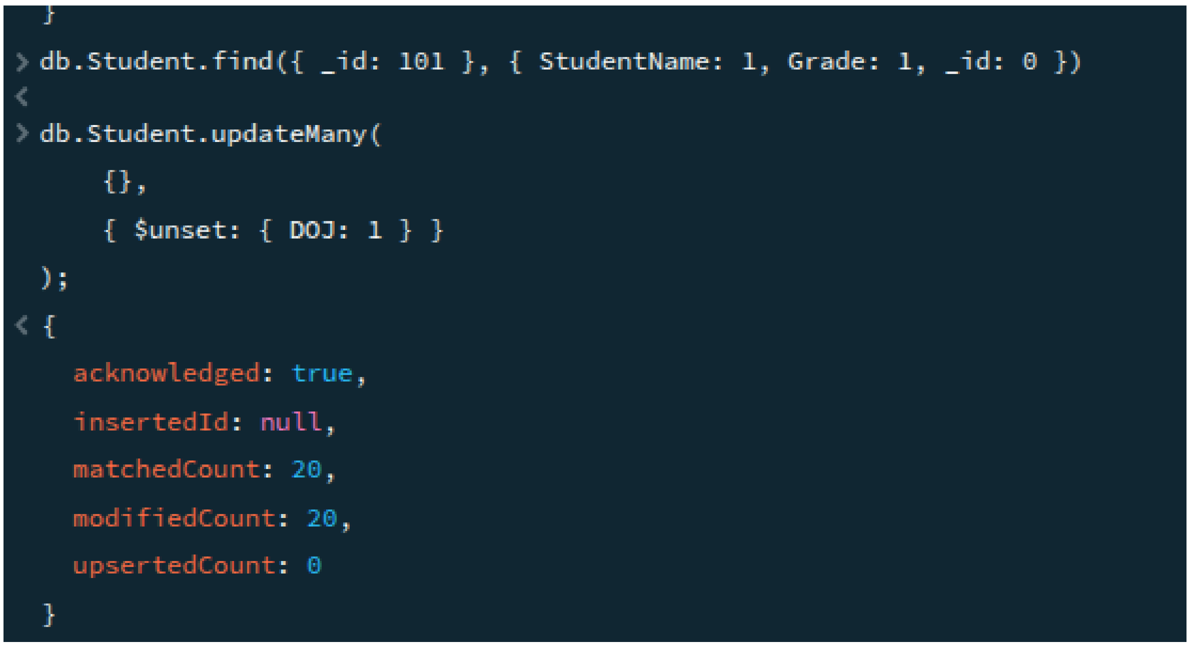
Find all documents in proper format. (Without _Id field)



```
> db.Student.find({}, { _id: 0 })
< {
    StudentRollNo: 101,
    StudentName: 'Ajay Rathod',
    Grade: 'VIII',
    Hobbies: [
      'Football',
      'Reading'
    ],
    DOJ: 2024-08-01T00:00:00.000Z,
    Address: 'Royal Street'
  }
  {
```

Name : Parth Harpal Roll No: 24CI2110031

5. Remove ‘DOJ’ field in Student Collection.

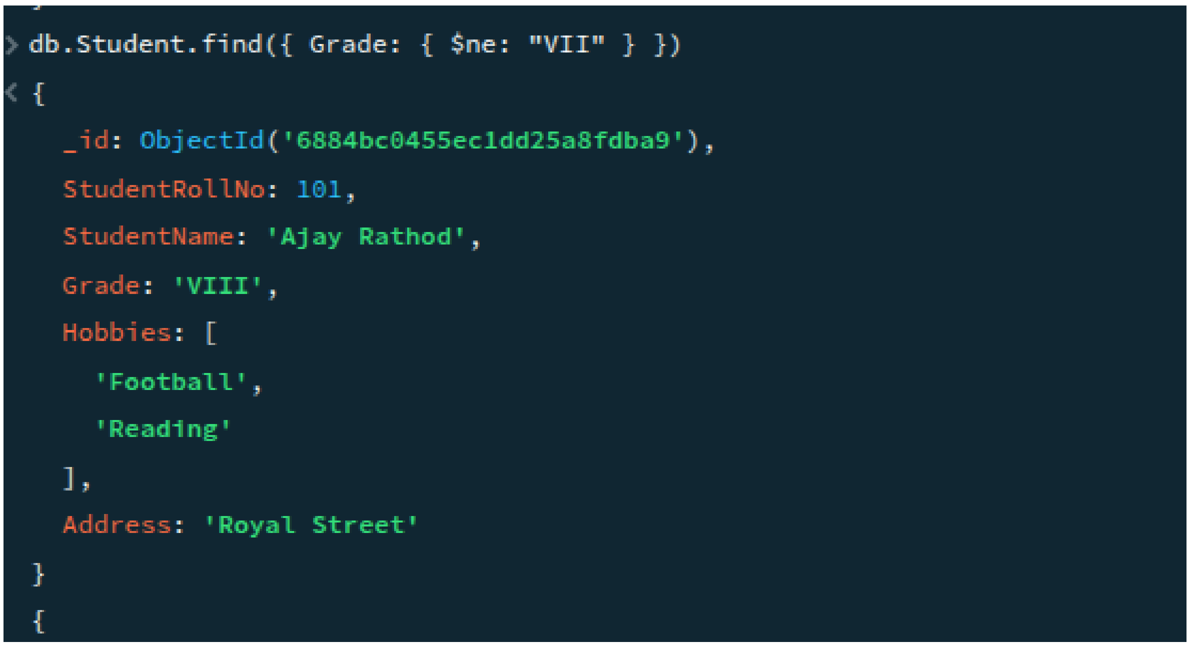


```
  }
> db.Student.find({ _id: 101 }, { StudentName: 1, Grade: 1, _id: 0 })
<
> db.Student.updateMany(
      {},
      { $unset: { DOJ: 1 } }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 20,
    modifiedCount: 20,
    upsertedCount: 0
  }
```

Name : Parth Harpal Roll No: 24CI2110031

6.Find those documents where the Grade is not set to ‘VII’.

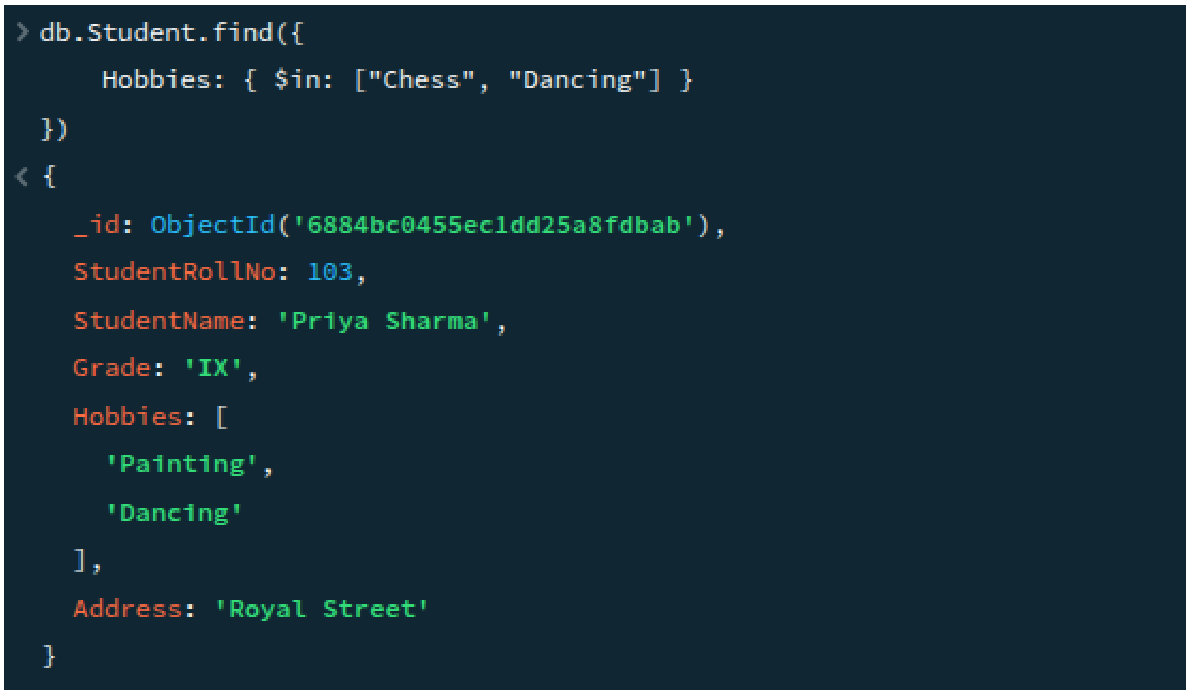
7.



```
> db.Student.find({ Grade: { $ne: "VII" } })
< {
    _id: ObjectId('6884bc0455ec1dd25a8fdba9'),
    StudentRollNo: 101,
    StudentName: 'Ajay Rathod',
    Grade: 'VIII',
    Hobbies: [
      'Football',
      'Reading'
    ],
    Address: 'Royal Street'
  }
  {
```

Find those documents where the Hobbies is set to either ‘Chess’ or is set to ‘Dancing”.

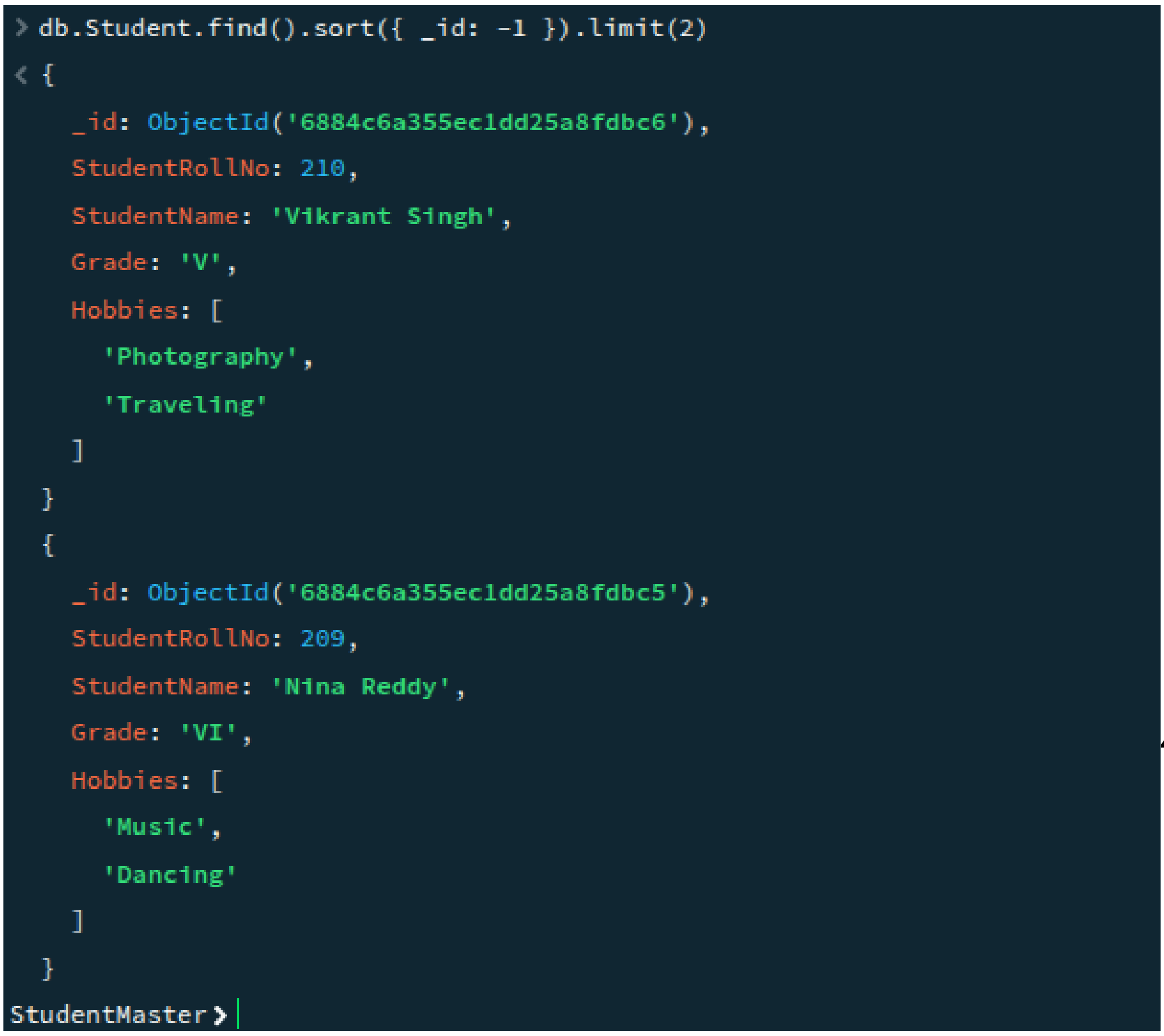
8.



```
> db.Student.find({
      Hobbies: { $in: ["Chess", "Dancing"] }
  })
< {
    _id: ObjectId('6884bc0455ec1dd25a8fdbab'),
    StudentRollNo: 103,
    StudentName: 'Priya Sharma',
    Grade: 'IX',
    Hobbies: [
      'Painting',
      'Dancing'
    ],
    Address: 'Royal Street'
  }
```

Display the last two records.

Name : Parth Harpal Roll No: 24CI2110031



```
> db.Student.find().sort({ _id: -1 }).limit(2)
< {
    _id: ObjectId('6884c6a355ec1dd25a8fdbc6'),
    StudentRollNo: 210,
    StudentName: 'Vikrant Singh',
    Grade: 'V',
    Hobbies: [
      'Photography',
      'Traveling'
    ]
  }
  {
    _id: ObjectId('6884c6a355ec1dd25a8fdbc5'),
    StudentRollNo: 209,
    StudentName: 'Nina Reddy',
    Grade: 'VI',
    Hobbies: [
      'Music',
      'Dancing'
    ]
  }
StudentMaster >
```

Q- 4)

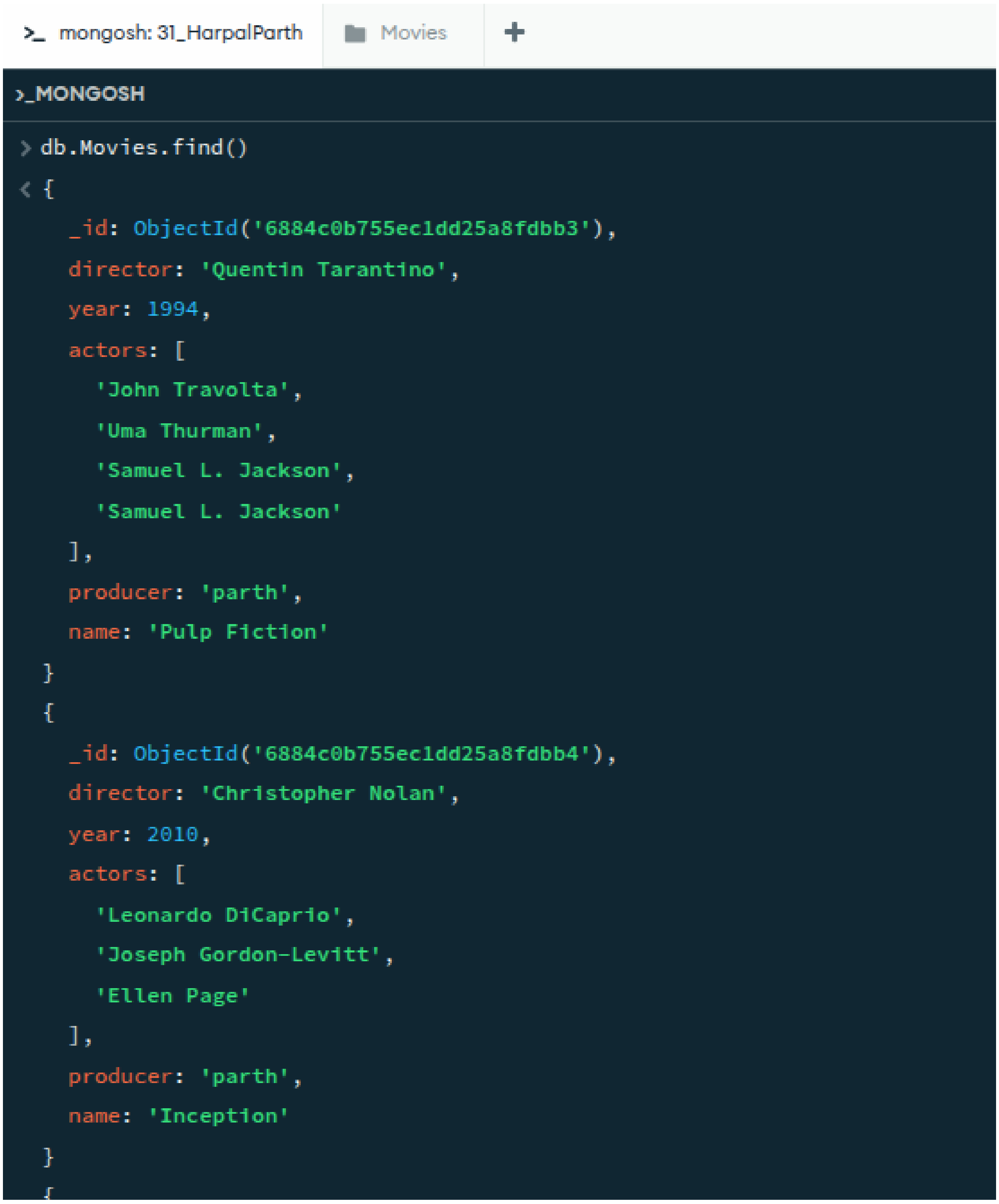
Create a MovieMaker Database with a collection called “Movies “containing documents with some or all of the following fields: titles, directors, years, actors.

Perform the following operations on the database:

1. Retrieve all documents

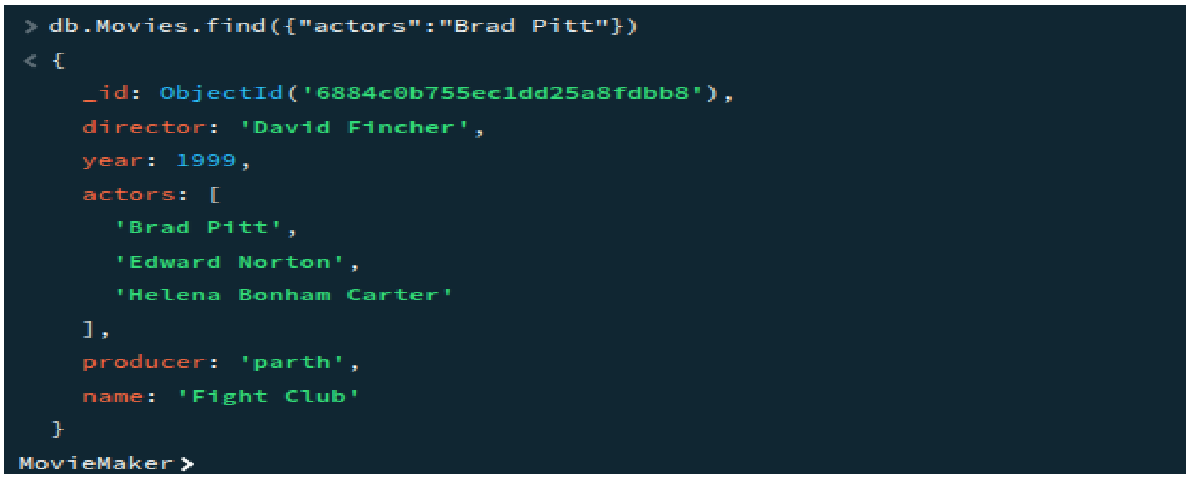
Name : Parth Harpal Roll No: 24CI2110031

2.



```
> db.Movies.find()
< {
    _id: ObjectId('6884c0b755ec1dd25a8fdbb3'),
    director: 'Quentin Tarantino',
    year: 1994,
    actors: [
      'John Travolta',
      'Uma Thurman',
      'Samuel L. Jackson',
      'Samuel L. Jackson'
    ],
    producer: 'parth',
    name: 'Pulp Fiction'
  }
  {
    _id: ObjectId('6884c0b755ec1dd25a8fdbb4'),
    director: 'Christopher Nolan',
    year: 2010,
    actors: [
      'Leonardo DiCaprio',
      'Joseph Gordon-Levitt',
      'Ellen Page'
    ],
    producer: 'parth',
    name: 'Inception'
  }
  {
```

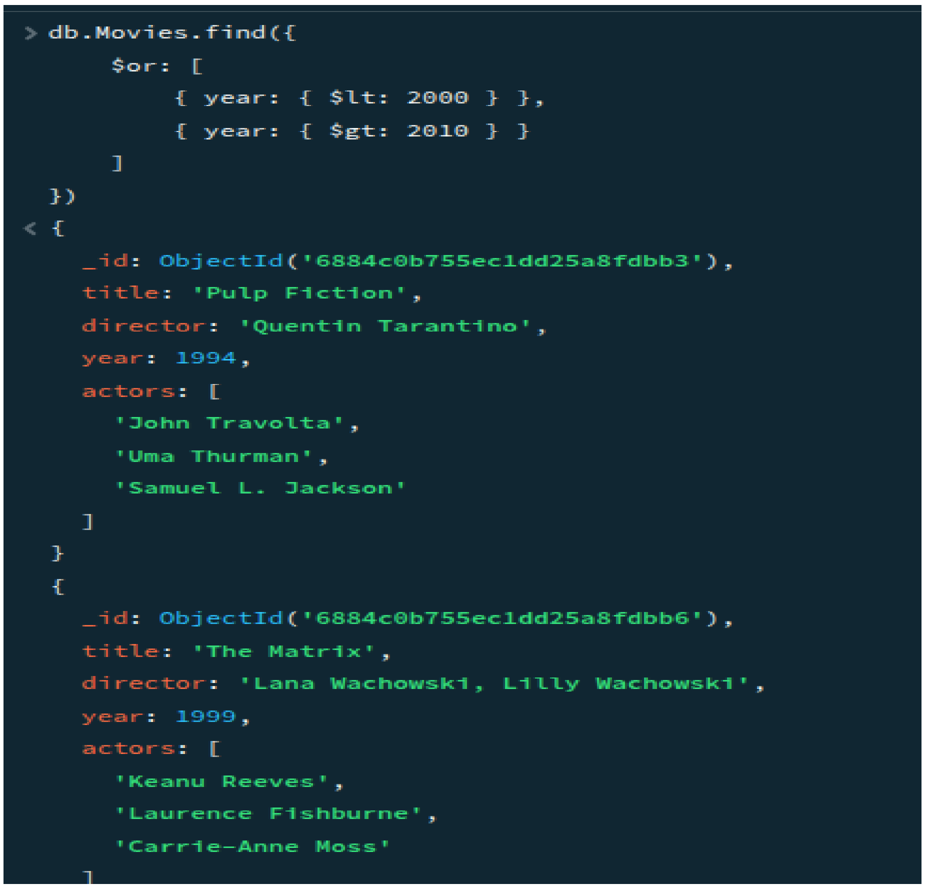
Retrieve all documents where actors include Brad Pitt.



```
> db.Movies.find({"actors":"Brad Pitt"})
< {
    _id: ObjectId('6884c0b755ec1dd25a8fdbb8'),
    director: 'David Fincher',
    year: 1999,
    actors: [
      'Brad Pitt',
      'Edward Norton',
      'Helena Bonham Carter'
    ],
    producer: 'parth',
    name: 'Fight Club'
  }
MovieMaker>
```

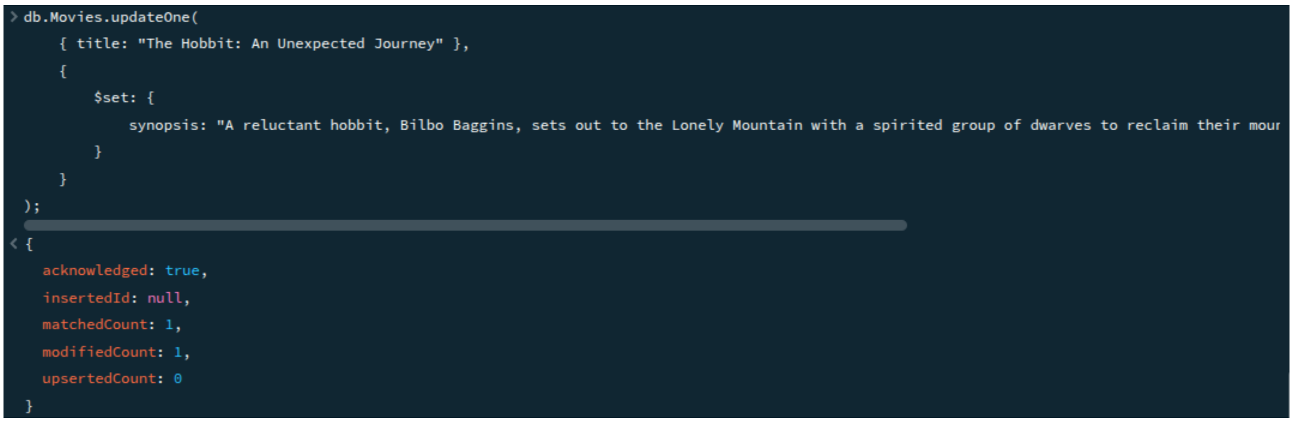
Name : Parth Harpal Roll No: 24CI2110031

3. Retrieve all movies released before the year 2000 or after 2010



```
> db.Movies.find({
      $or: [
          { year: { $lt: 2000 } },
          { year: { $gt: 2010 } }
      ]
  })
< {
    _id: ObjectId('6884c0b755ec1dd25a8fdbb3'),
    title: 'Pulp Fiction',
    director: 'Quentin Tarantino',
    year: 1994,
    actors: [
      'John Travolta',
      'Uma Thurman',
      'Samuel L. Jackson'
    ]
  }
  {
    _id: ObjectId('6884c0b755ec1dd25a8fdbb6'),
    title: 'The Matrix',
    director: 'Lana Wachowski, Lilly Wachowski',
    year: 1999,
    actors: [
      'Keanu Reeves',
      'Laurence Fishburne',
      'Carrie-Anne Moss'
    ]
```

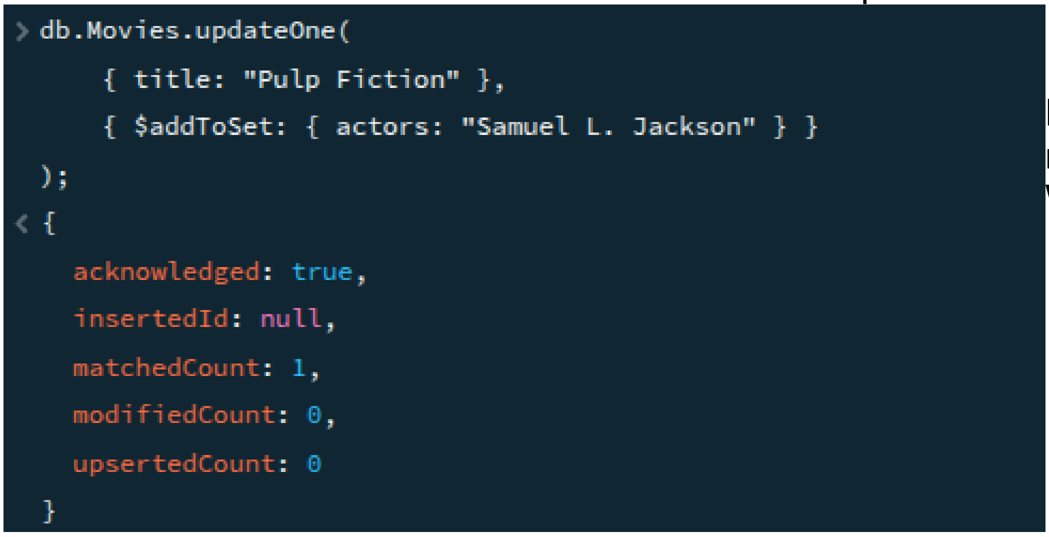
4. Add a synopsis “A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug.Where title is The Hobbit: An Unexpected Journey”.



```
> db.Movies.updateOne(
      { title: "The Hobbit: An Unexpected Journey" },
      {
          $set: {
              synopsis: "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mour
          }
      }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
```

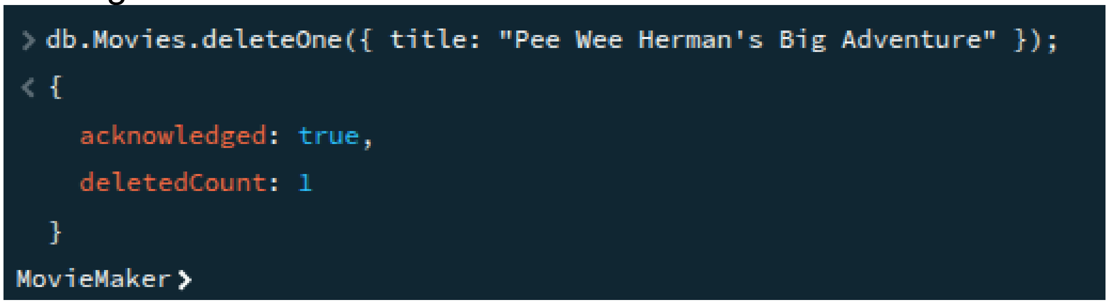
Name : Parth Harpal Roll No: 24CI2110031

5. Add an actor named Samuel L. Jackson to the movie Pulp Fiction



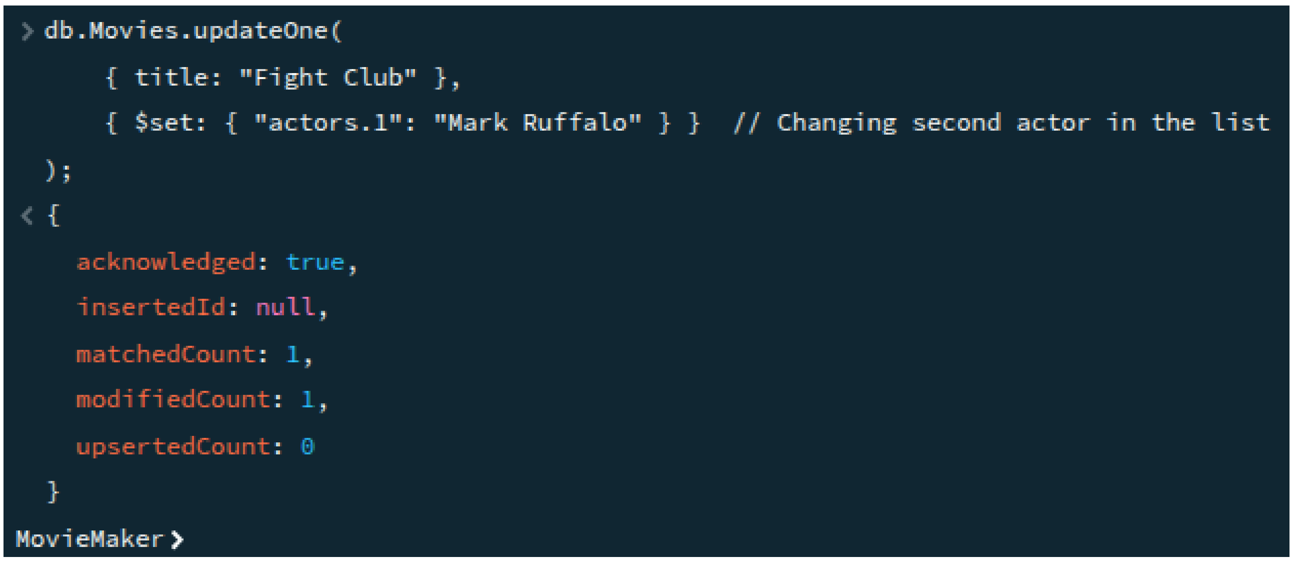
```
> db.Movies.updateOne(
      { title: "Pulp Fiction" },
      { $addToSet: { actors: "Samuel L. Jackson" } }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 0,
    upsertedCount: 0
  }
```

6. Delete the movie Pee Wee Hermans Big Adventure”



```
> db.Movies.deleteOne({ title: "Pee Wee Herman's Big Adventure" });
< {
    acknowledged: true,
    deletedCount: 1
  }
MovieMaker >
```

7. Update one of the actors in any given movie title.



```
> db.Movies.updateOne(
      { title: "Fight Club" },
      { $set: { "actors.1": "Mark Ruffalo" } }  // Changing second actor in the list
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
MovieMaker >
```

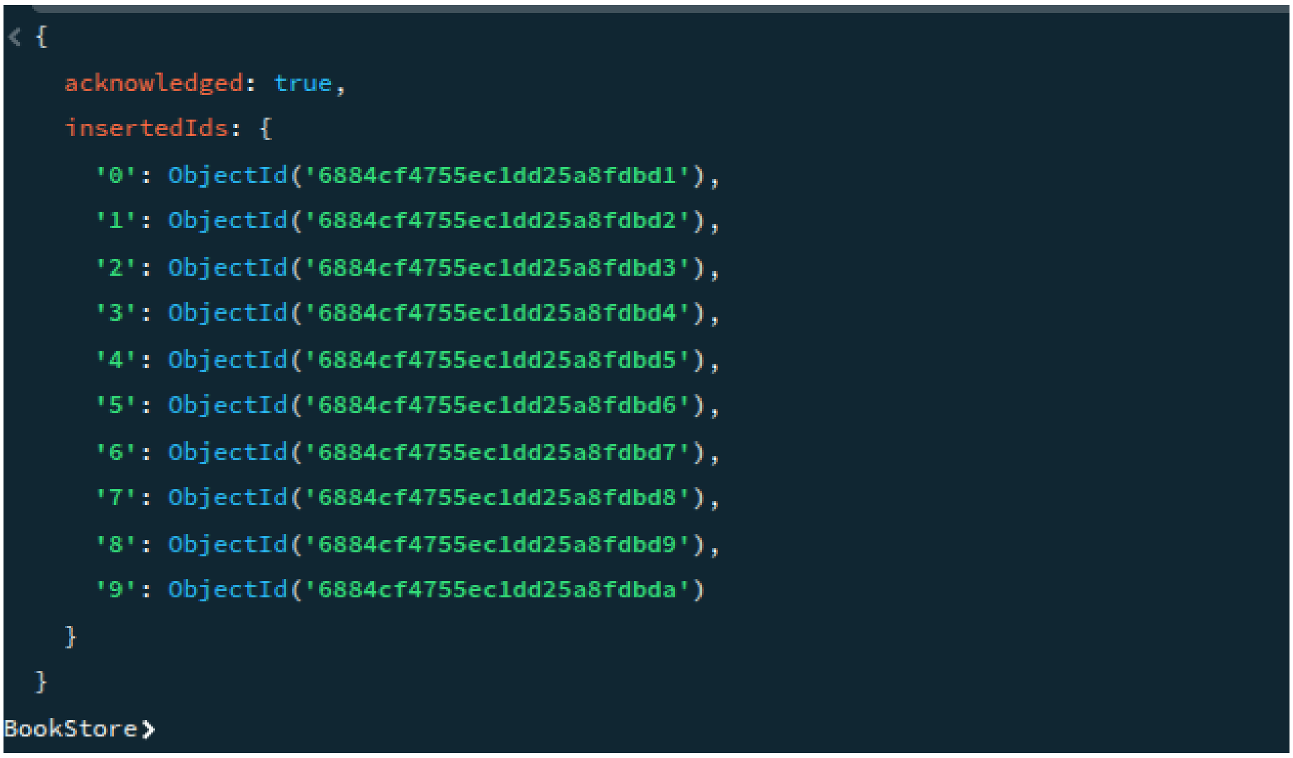
Name : Parth Harpal Roll No: 24CI2110031

Q-5) Create a database named “BookStore” in MongoDB with acollection called “Books” containing documents with some or all of the following fields: bookId, bookTitle, authors(containing fields: authorName),publicationYear, publisher, Orders(containing fields: OrderedId, orderDate, customerName, price, quantityOrdered, discount).

Note that a book may have one or more authors and orders.
Also, the same OrdereId can be present in one or more books.
Perform the following operations on the database:

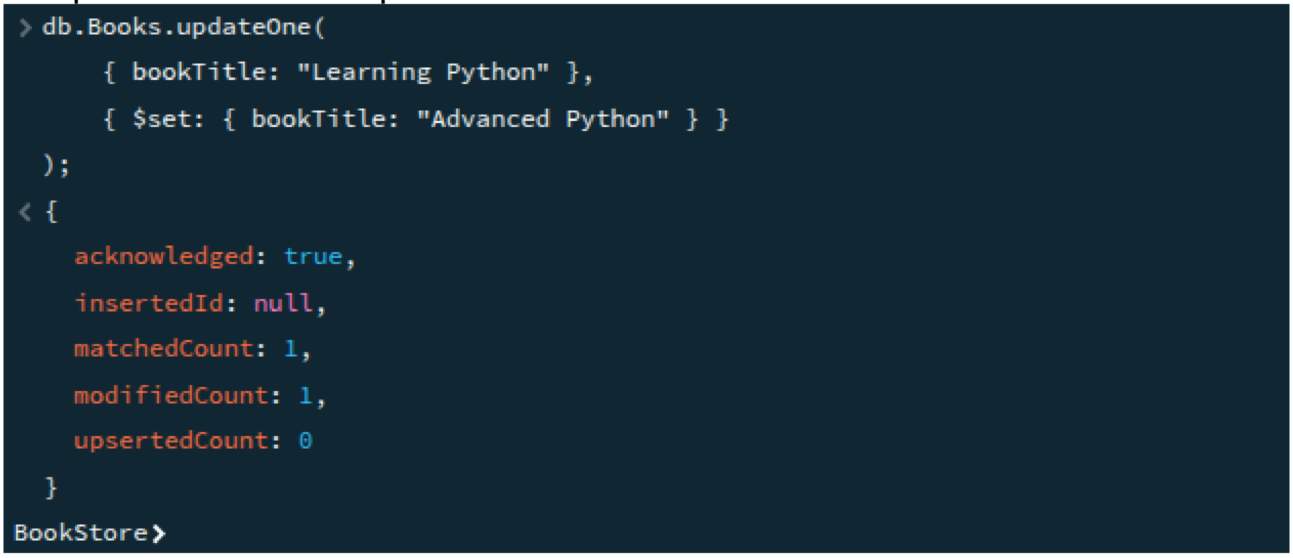
1. Insert records for 10 books from 5 authors, and at least 20 orders in total.

Name : Parth Harpal   Roll No: 24CI2110031



```
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('6884cf4755ec1dd25a8fdbd1'),
      '1': ObjectId('6884cf4755ec1dd25a8fdbd2'),
      '2': ObjectId('6884cf4755ec1dd25a8fdbd3'),
      '3': ObjectId('6884cf4755ec1dd25a8fdbd4'),
      '4': ObjectId('6884cf4755ec1dd25a8fdbd5'),
      '5': ObjectId('6884cf4755ec1dd25a8fdbd6'),
      '6': ObjectId('6884cf4755ec1dd25a8fdbd7'),
      '7': ObjectId('6884cf4755ec1dd25a8fdbd8'),
      '8': ObjectId('6884cf4755ec1dd25a8fdbd9'),
      '9': ObjectId('6884cf4755ec1dd25a8fdbda')
    }
  }
BookStore>
```

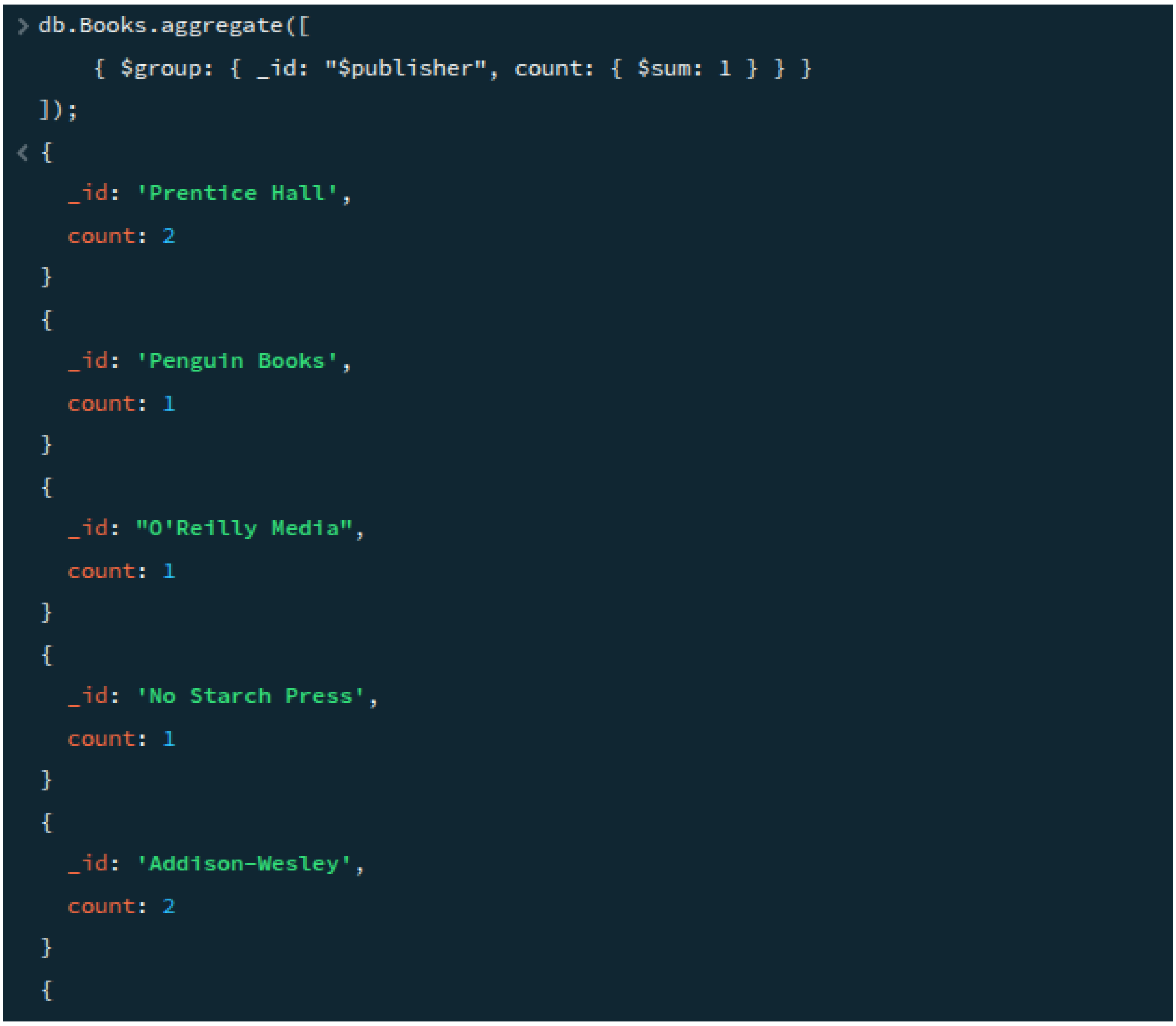
2. Update the title of a particular book.



```
> db.Books.updateOne(
      { bookTitle: "Learning Python" },
      { $set: { bookTitle: "Advanced Python" } }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
BookStore>
```

4. Display the number of books from each publisher.

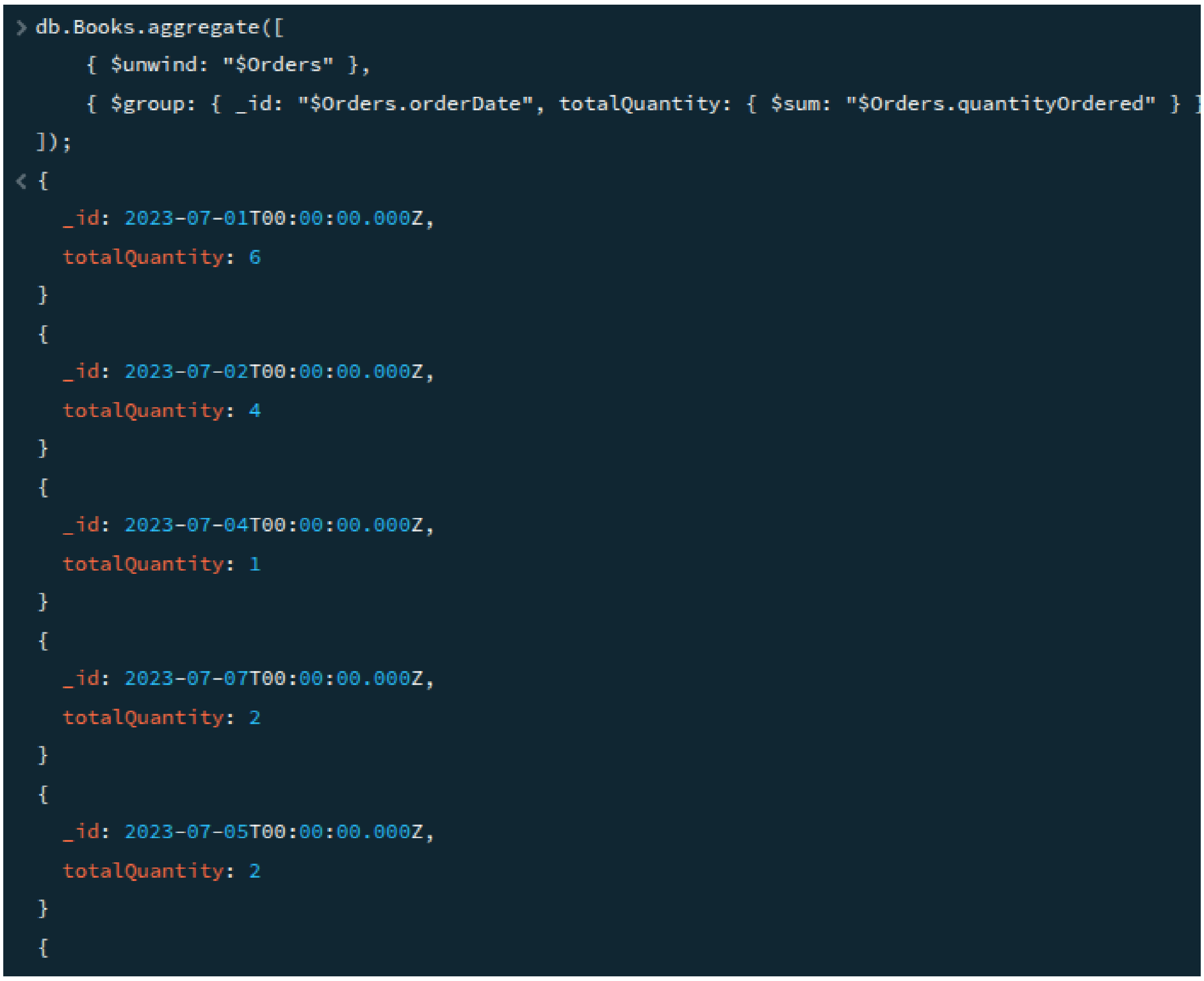
Name : Parth Harpal Roll No: 24CI2110031



```
> db.Books.aggregate([
      { $group: { _id: "$publisher", count: { $sum: 1 } } }
  ]);
< {
    _id: 'Prentice Hall',
    count: 2
  }
  {
    _id: 'Penguin Books',
    count: 1
  }
  {
    _id: "O'Reilly Media",
    count: 1
  }
  {
    _id: 'No Starch Press',
    count: 1
  }
  {
    _id: 'Addison-Wesley',
    count: 2
  }
  {
```

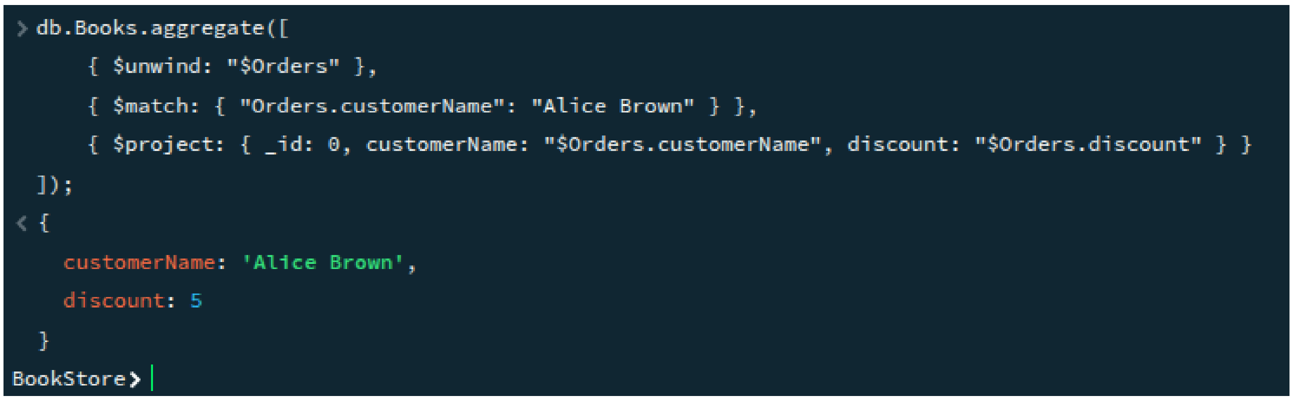
5. Display the total quantity of books ordered for each date.

Name : Parth Harpal	Roll No: 24CI2110031



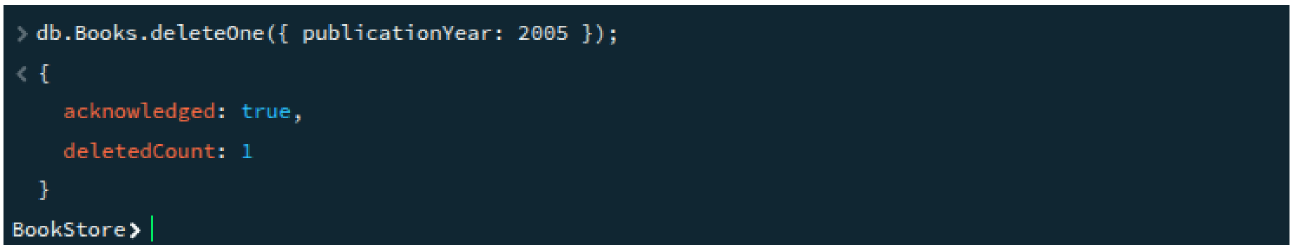
```
> db.Books.aggregate([
      { $unwind: "$Orders" },
      { $group: { _id: "$Orders.orderDate", totalQuantity: { $sum: "$Orders.quantityOrdered" } 
  ]);
< {
    _id: 2023-07-01T00:00:00.000Z,
    totalQuantity: 6
  }
  {
    _id: 2023-07-02T00:00:00.000Z,
    totalQuantity: 4
  }
  {
    _id: 2023-07-04T00:00:00.000Z,
    totalQuantity: 1
  }
  {
    _id: 2023-07-07T00:00:00.000Z,
    totalQuantity: 2
  }
  {
    _id: 2023-07-05T00:00:00.000Z,
    totalQuantity: 2
  }
  {
```

6. Display the discount offered to a particular customer.



```
> db.Books.aggregate([
      { $unwind: "$Orders" },
      { $match: { "Orders.customerName": "Alice Brown" } },
      { $project: { _id: 0, customerName: "$Orders.customerName", discount: "$Orders.discount" } }
  ]);
< {
    customerName: 'Alice Brown',
    discount: 5
  }
BookStore>
```

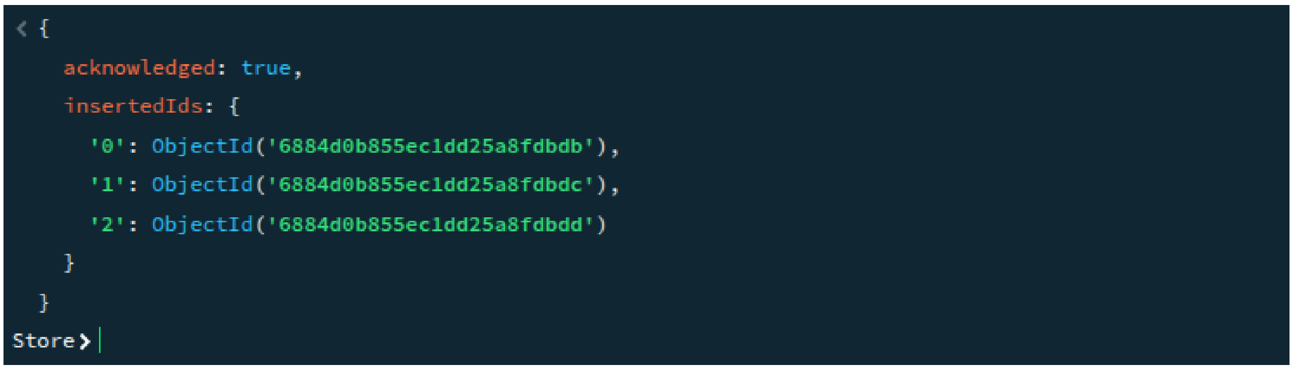
7. Delete the book having particular publicationYear.



```
> db.Books.deleteOne({ publicationYear: 2005 });
< {
    acknowledged: true,
    deletedCount: 1
  }
BookStore>
```

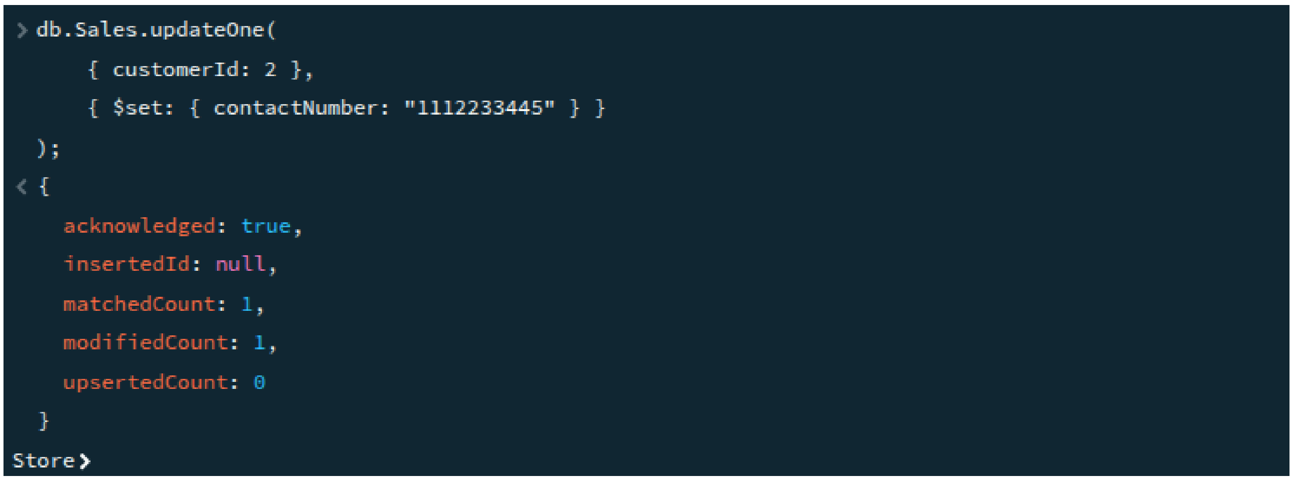
Q-6) Create a database named “Store” in MongoDB with a collection called “Sales” containing documents with following fields: customerId, customerName, gender, dataOfBirth, contactNumber, address (containing fields: houseNo, street, area, city, pincode), orders(containing fields: orderId, orderDate, items(containing fields: itemId, itemName, itemPrice, quantityOrdered, discount)). Note that some customers may not provide their date of birth and/or contact number. Also, not all products would be sold at a discount. Perform the following operations on the database:

1. Insert records for 3 customers and 5 items in at least 20 orders.



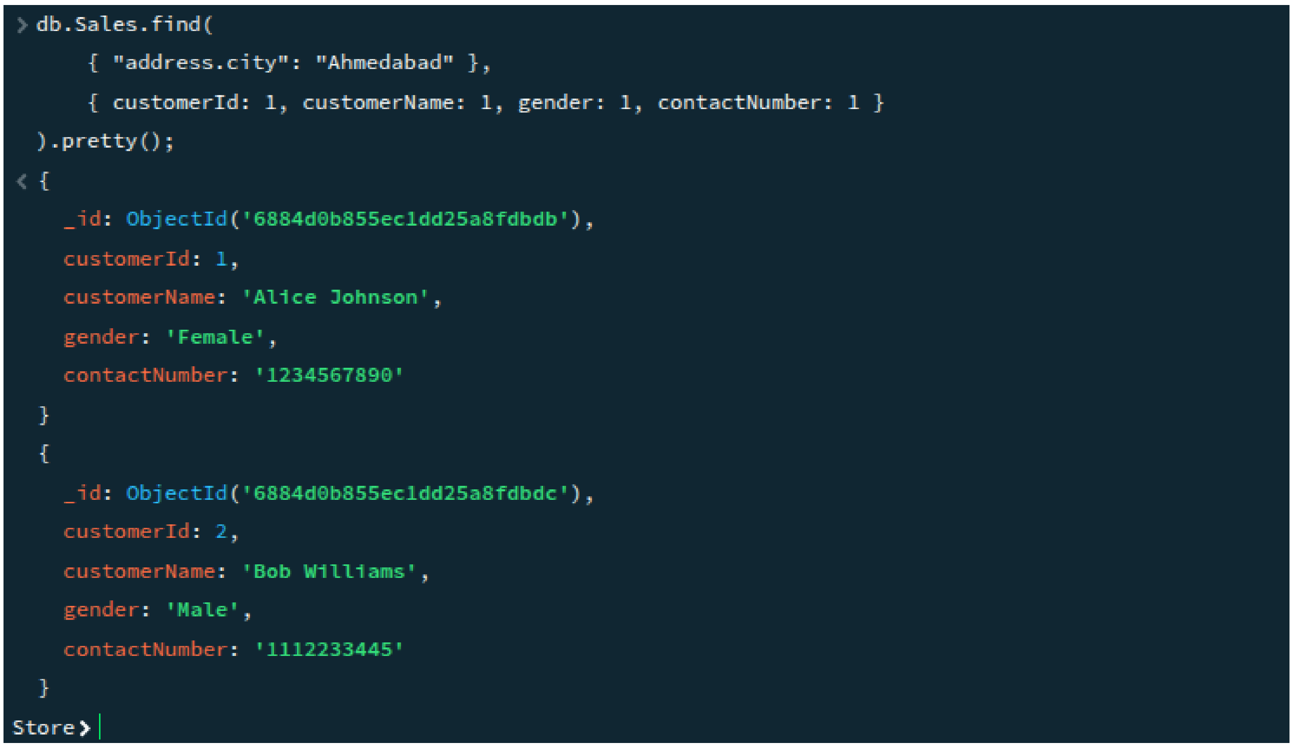
```
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('6884d0b855ec1dd25a8fdbdb'),
      '1': ObjectId('6884d0b855ec1dd25a8fdbdc'),
      '2': ObjectId('6884d0b855ec1dd25a8fdbdd')
    }
  }
Store>
```

2. Update the contact number of a particular customer.



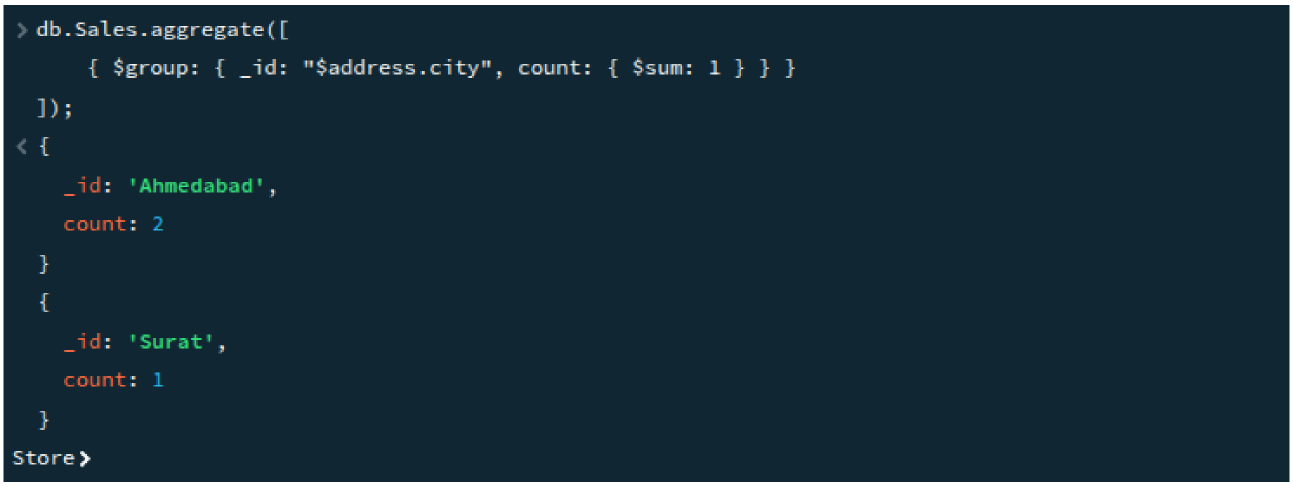
```
> db.Sales.updateOne(
      { customerId: 2 },
      { $set: { contactNumber: "1112233445" } }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
Store>
```

3. Display customerId, customerName, gender, contactNumber, of customers residing in “Ahmedabad”.



```
> db.Sales.find(
      { "address.city": "Ahmedabad" },
      { customerId: 1, customerName: 1, gender: 1, contactNumber: 1 }
  ).pretty();
< {
    _id: ObjectId('6884d0b855ec1dd25a8fdbdb'),
    customerId: 1,
    customerName: 'Alice Johnson',
    gender: 'Female',
    contactNumber: '1234567890'
  }
  {
    _id: ObjectId('6884d0b855ec1dd25a8fdbdc'),
    customerId: 2,
    customerName: 'Bob Williams',
    gender: 'Male',
    contactNumber: '1112233445'
  }
Store>
```

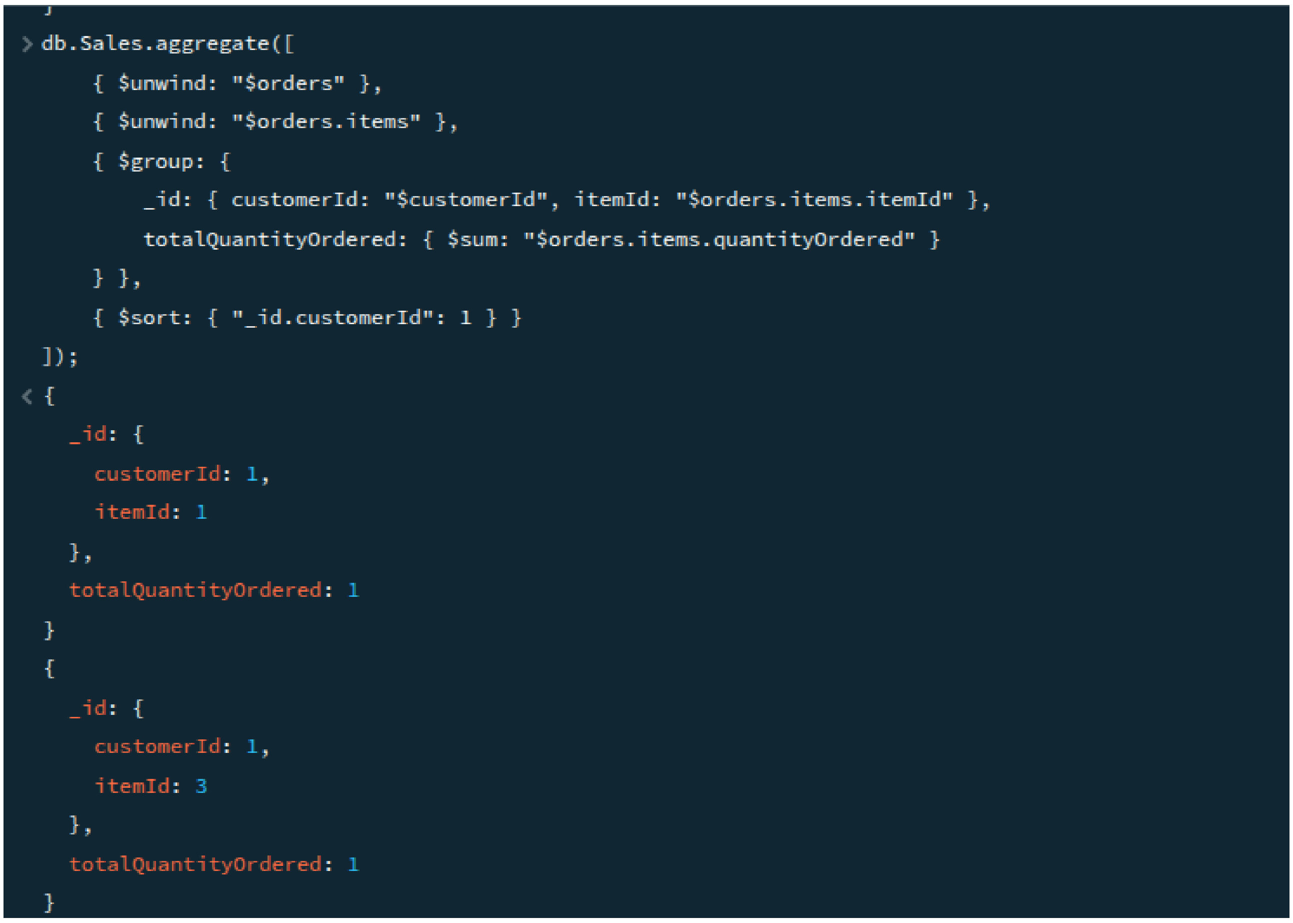
4. Display city-wise count of customers



```
> db.Sales.aggregate([
      { $group: { _id: "$address.city", count: { $sum: 1 } } }
  ]);
< {
    _id: 'Ahmedabad',
    count: 2
  }
  {
    _id: 'Surat',
    count: 1
  }
Store>
```

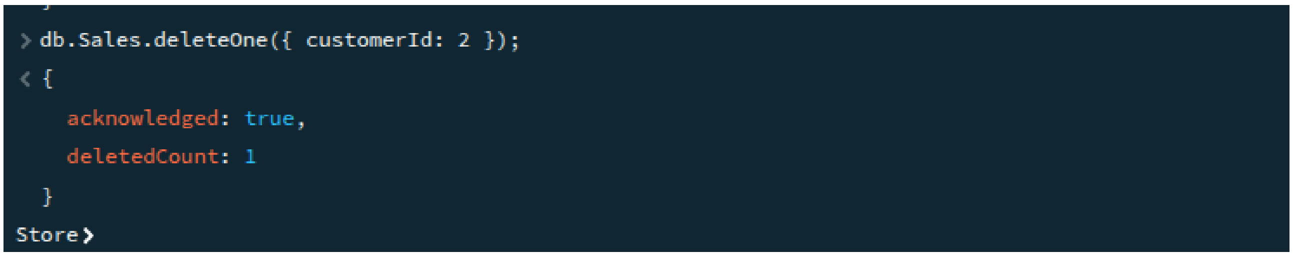
Name : Parth Harpal Roll No: 24CI2110031

5. Display total quantity ordered of items for each customer.



```
> db.Sales.aggregate([
      { $unwind: "$orders" },
      { $unwind: "$orders.items" },
      { $group: {
          _id: { customerId: "$customerId", itemId: "$orders.items.itemId" },
          totalQuantityOrdered: { $sum: "$orders.items.quantityOrdered" }
      } },
      { $sort: { "_id.customerId": 1 } }
  ]);
< {
    _id: {
      customerId: 1,
      itemId: 1
    },
    totalQuantityOrdered: 1
  }
  {
    _id: {
      customerId: 1,
      itemId: 3
    },
    totalQuantityOrdered: 1
  }
```

6. Delete the Customer with specific customerId.

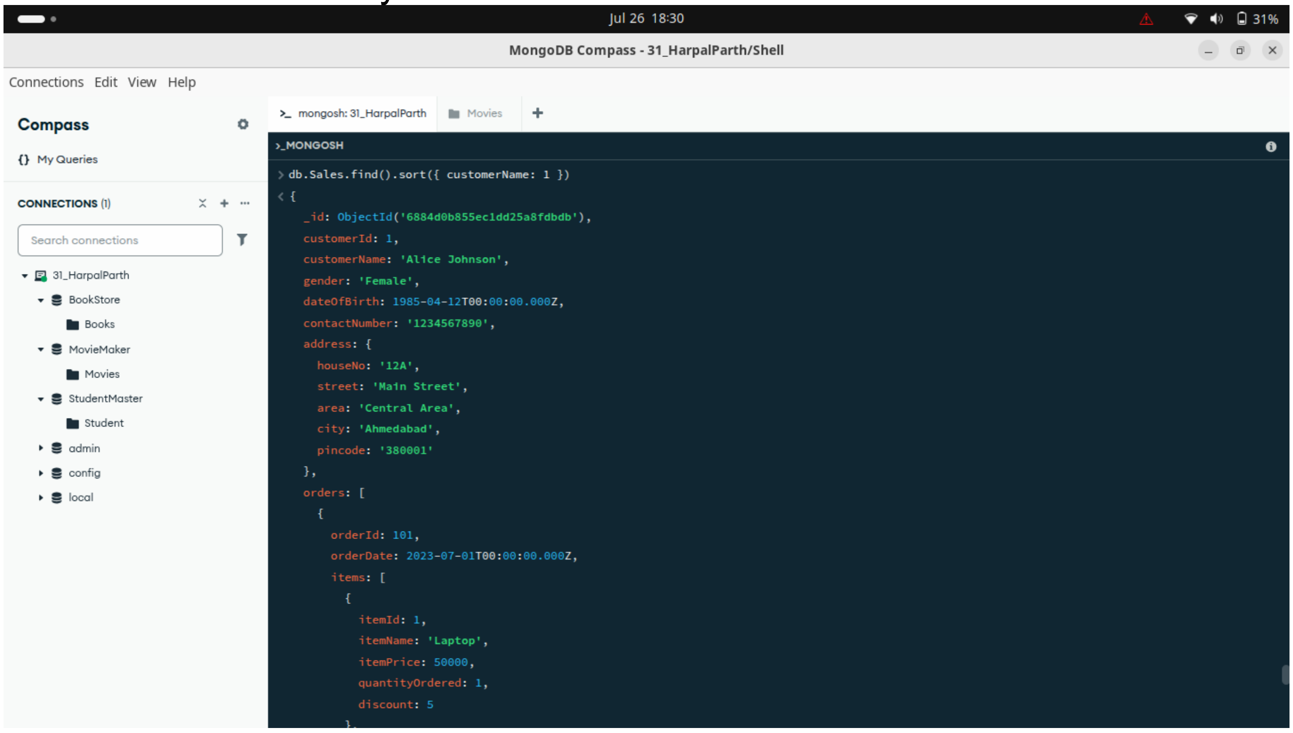


```
> db.Sales.deleteOne({ customerId: 2 });
< {
    acknowledged: true,
    deletedCount: 1
  }
Store>
```

Name : Parth Harpal

Roll No: 24CI2110031

7. Sort the Customer by customerName.



```
> db.Sales.find().sort({ customerName: 1 })
< {
    _id: ObjectId('6884d0b855ec1dd25a8fdbdb'),
    customerId: 1,
    customerName: 'Alice Johnson',
    gender: 'Female',
    dateOfBirth: 1985-04-12T00:00:00.000Z,
    contactNumber: '1234567890',
    address: {
      houseNo: '12A',
      street: 'Main Street',
      area: 'Central Area',
      city: 'Ahmedabad',
      pincode: '380001'
    },
    orders: [
      {
        orderId: 101,
        orderDate: 2023-07-01T00:00:00.000Z,
        items: [
          {
            itemId: 1,
            itemName: 'Laptop',
            itemPrice: 50000,
            quantityOrdered: 1,
            discount: 5
```