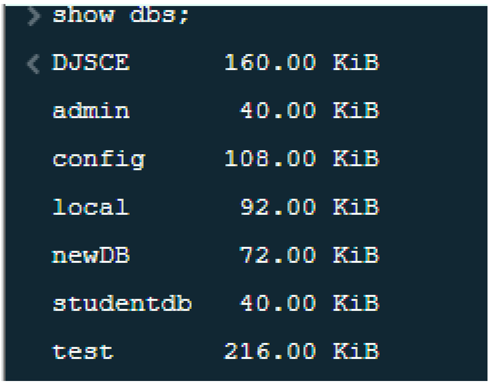
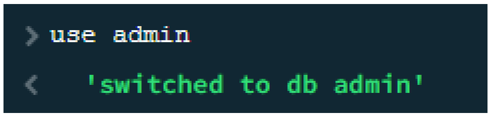
1. Show existing databases.



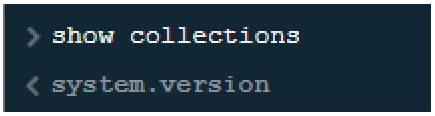
```
> show dbs;
< DJSCE       160.00 KiB
  admin        40.00 KiB
  config      108.00 KiB
  local        92.00 KiB
  newDB        72.00 KiB
  studentdb    40.00 KiB
  test        216.00 KiB
```

2. Use one the database.



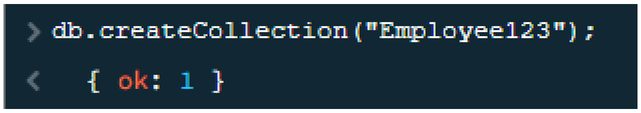
```
> use admin
<   'switched to db admin'
```

3. Show collections in the above database, say example: admin



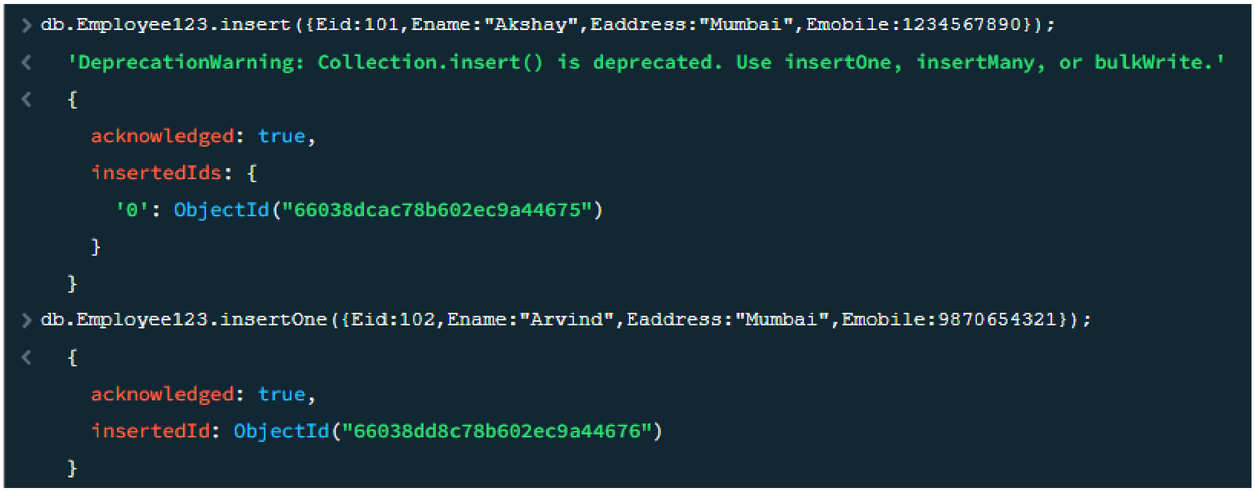
```
> show collections
< system.version
```

4. Create a new collection say, Employee. (Create as Employee with last 3 digits of SAPID, eg. Employee123)



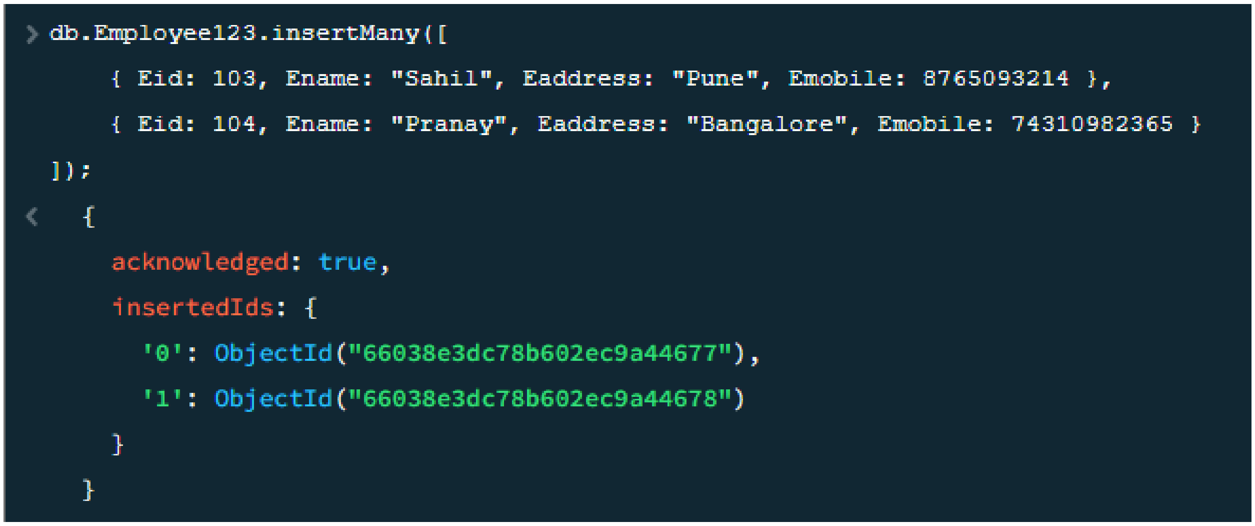
```
> db.createCollection("Employee123");
<   { ok: 1 }
```

5. Insert one tuple in the collection as:



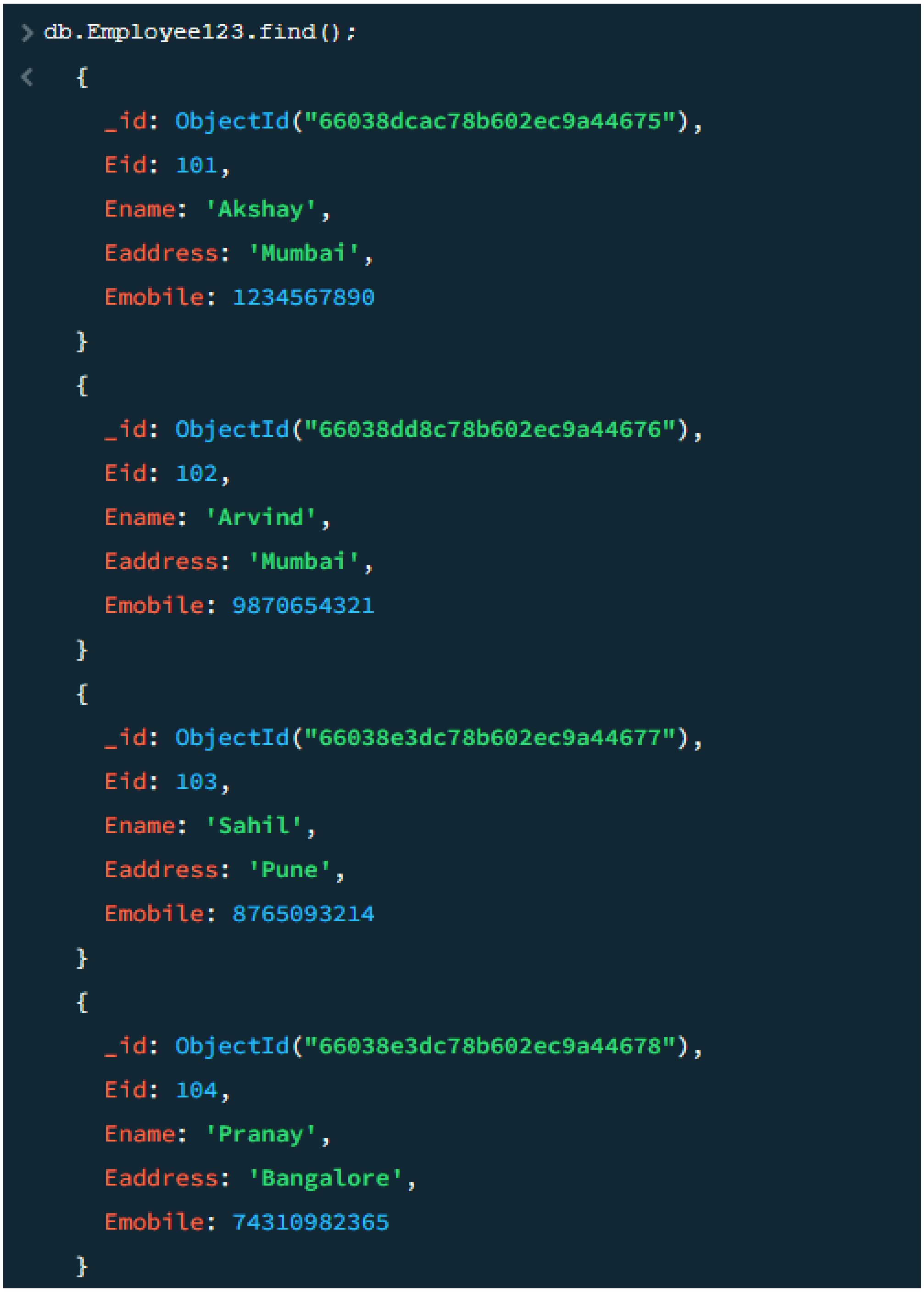
```
> db.Employee123.insert({Eid:101,Ename:"Akshay",Eaddress:"Mumbai",Emobile:1234567890});
<   'DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.'
<   {
      acknowledged: true,
      insertedIds: {
        '0': ObjectId("66038dcac78b602ec9a44675")
      }
    }
> db.Employee123.insertOne({Eid:102,Ename:"Arvind",Eaddress:"Mumbai",Emobile:9870654321});
<   {
      acknowledged: true,
      insertedId: ObjectId("66038dd8c78b602ec9a44676")
    }
```

6. Insert multiple tuples in the collection as:



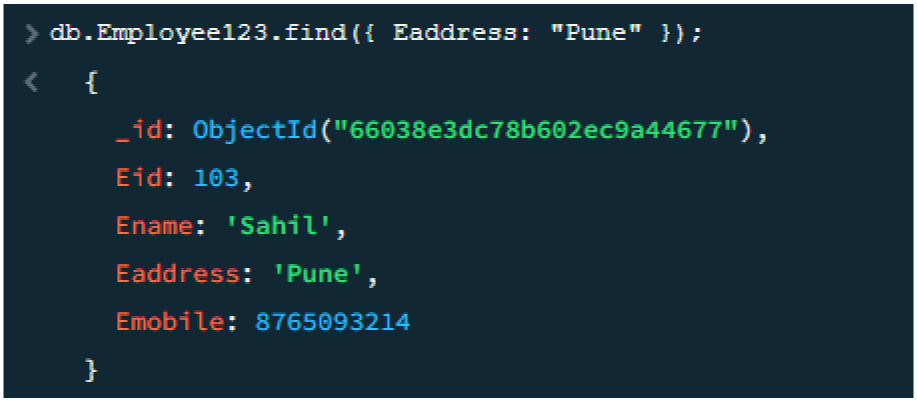
```
> db.Employee123.insertMany([
      { Eid: 103, Ename: "Sahil", Eaddress: "Pune", Emobile: 8765093214 },
      { Eid: 104, Ename: "Pranay", Eaddress: "Bangalore", Emobile: 74310982365 }
  ]);
<   {
      acknowledged: true,
      insertedIds: {
        '0': ObjectId("66038e3dc78b602ec9a44677"),
        '1': ObjectId("66038e3dc78b602ec9a44678")
      }
    }
```

7. Display the contents of the collection.



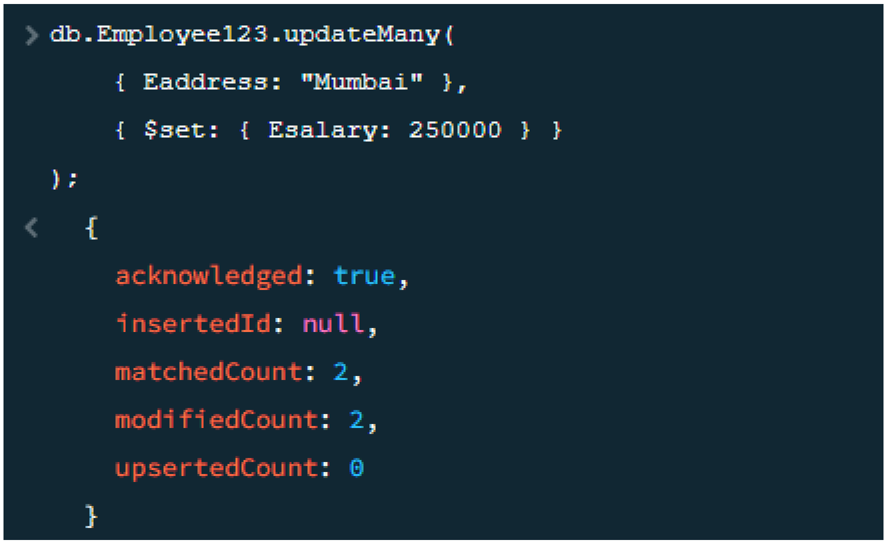
```
> db.Employee123.find();
<   {
      _id: ObjectId("66038dcac78b602ec9a44675"),
      Eid: 101,
      Ename: 'Akshay',
      Eaddress: 'Mumbai',
      Emobile: 1234567890
    }
    {
      _id: ObjectId("66038dd8c78b602ec9a44676"),
      Eid: 102,
      Ename: 'Arvind',
      Eaddress: 'Mumbai',
      Emobile: 9870654321
    }
    {
      _id: ObjectId("66038e3dc78b602ec9a44677"),
      Eid: 103,
      Ename: 'Sahil',
      Eaddress: 'Pune',
      Emobile: 8765093214
    }
    {
      _id: ObjectId("66038e3dc78b602ec9a44678"),
      Eid: 104,
      Ename: 'Pranay',
      Eaddress: 'Bangalore',
      Emobile: 74310982365
    }
```

8. Display the contents of the collection with address Pune.



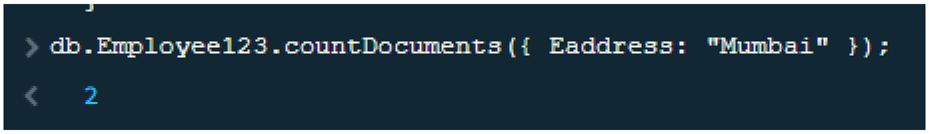
```
> db.Employee123.find({ Eaddress: "Pune" });
<   {
      _id: ObjectId("66038e3dc78b602ec9a44677"),
      Eid: 103,
      Ename: 'Sahil',
      Eaddress: 'Pune',
      Emobile: 8765093214
    }
```

9. Update the collection to add new attribute as Esalary in all the tuples.



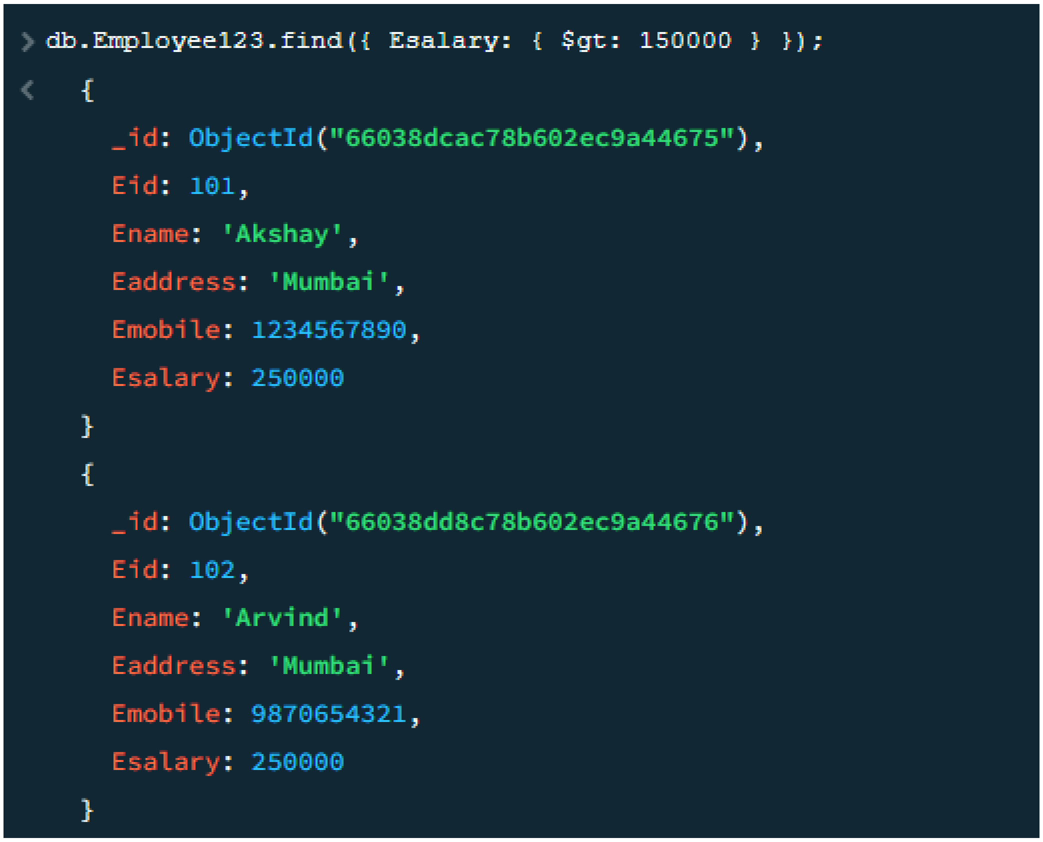
```
> db.Employee123.updateMany(
      { Eaddress: "Mumbai" },
      { $set: { Esalary: 250000 } }
  );
<   {
      acknowledged: true,
      insertedId: null,
      matchedCount: 2,
      modifiedCount: 2,
      upsertedCount: 0
    }
```

10. Count the entries with Eaddress as Mumbai.



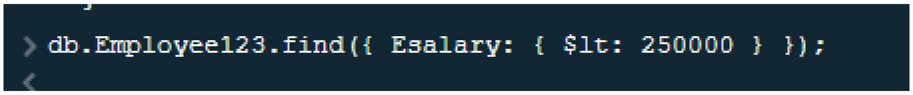
```
> db.Employee123.countDocuments({ Eaddress: "Mumbai" });
<   2
```

11. Display the details with Esalary greater than 150000.



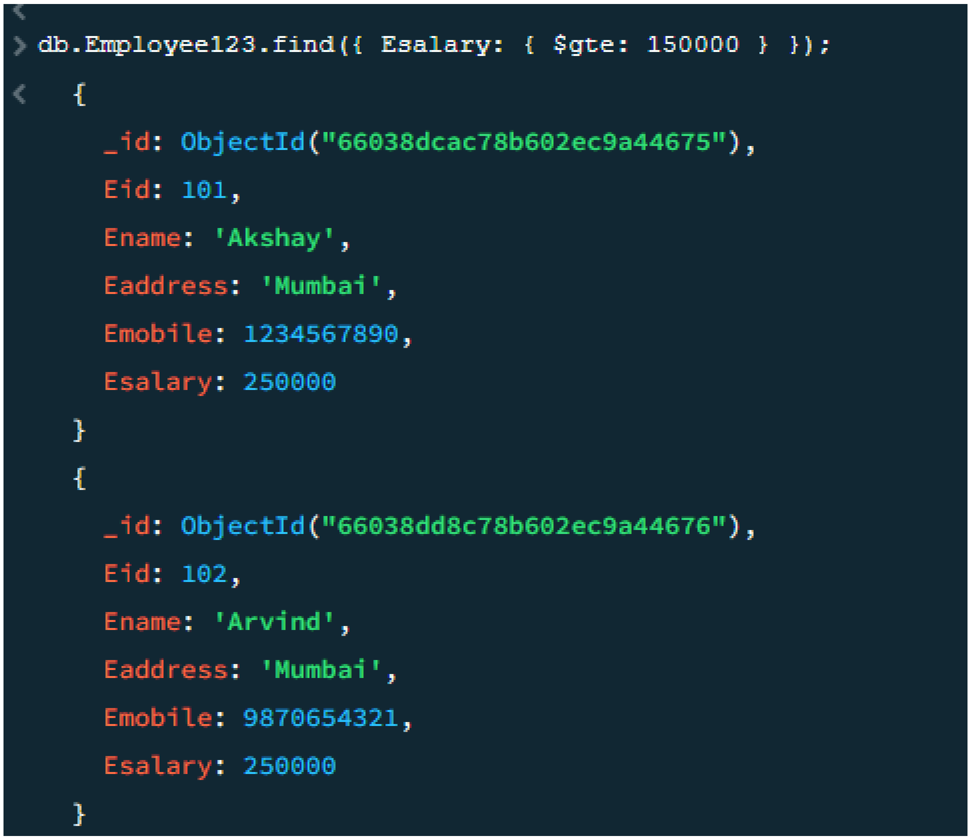
```
> db.Employee123.find({ Esalary: { $gt: 150000 } });
<   {
      _id: ObjectId("66038dcac78b602ec9a44675"),
      Eid: 101,
      Ename: 'Akshay',
      Eaddress: 'Mumbai',
      Emobile: 1234567890,
      Esalary: 250000
    }
    {
      _id: ObjectId("66038dd8c78b602ec9a44676"),
      Eid: 102,
      Ename: 'Arvind',
      Eaddress: 'Mumbai',
      Emobile: 9870654321,
      Esalary: 250000
    }
```

12. Display the details with Esalary lesser than 250000.



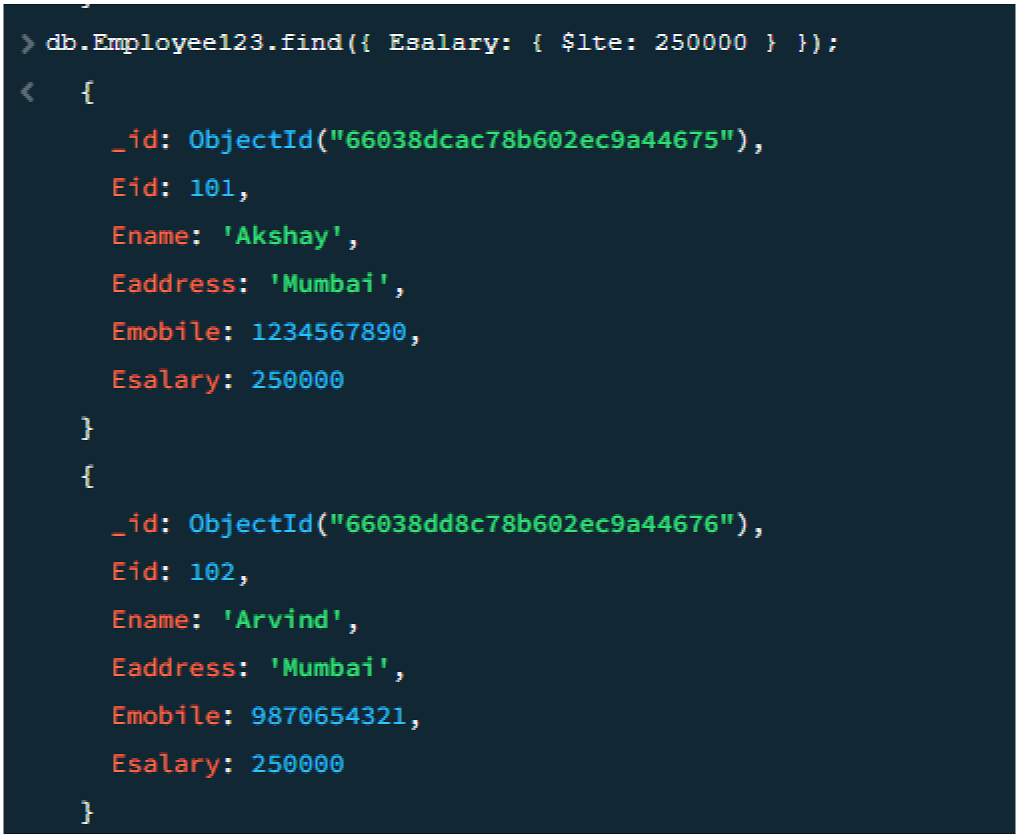
```
> db.Employee123.find({ Esalary: { $lt: 250000 } });
<
```

13. Display the details with Esalary greater than or equal to 150000.



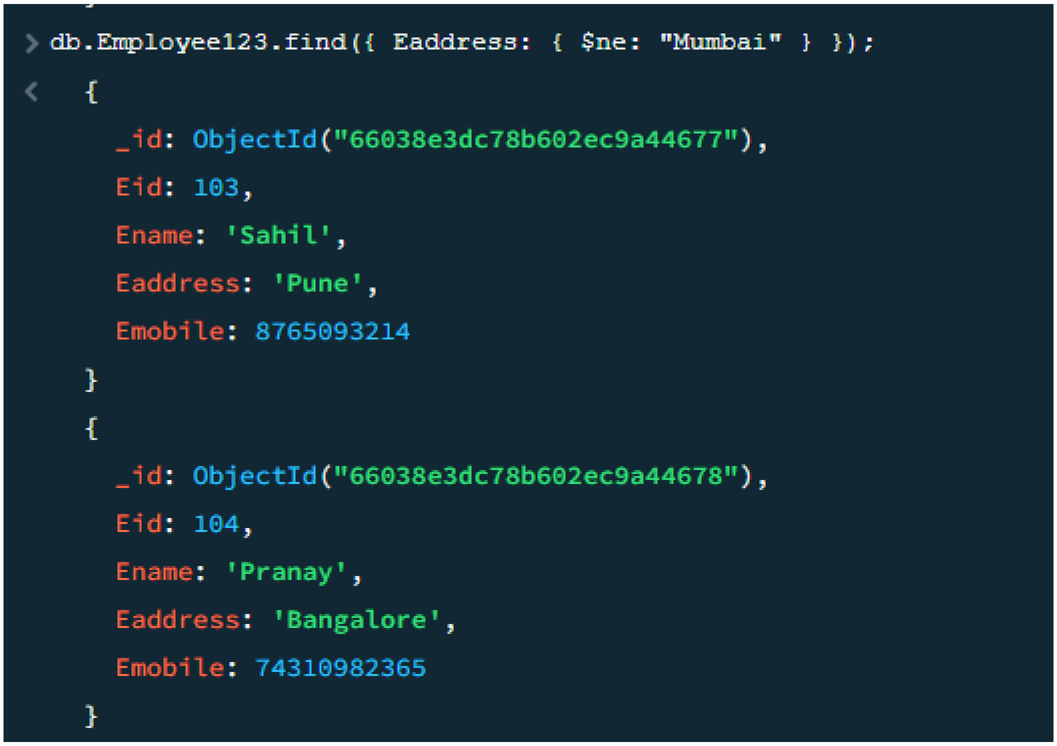
```
> db.Employee123.find({ Esalary: { $gte: 150000 } });
<   {
      _id: ObjectId("66038dcac78b602ec9a44675"),
      Eid: 101,
      Ename: 'Akshay',
      Eaddress: 'Mumbai',
      Emobile: 1234567890,
      Esalary: 250000
    }
    {
      _id: ObjectId("66038dd8c78b602ec9a44676"),
      Eid: 102,
      Ename: 'Arvind',
      Eaddress: 'Mumbai',
      Emobile: 9870654321,
      Esalary: 250000
    }
```

14. Display the details with Esalary lesser than or equal to 250000.



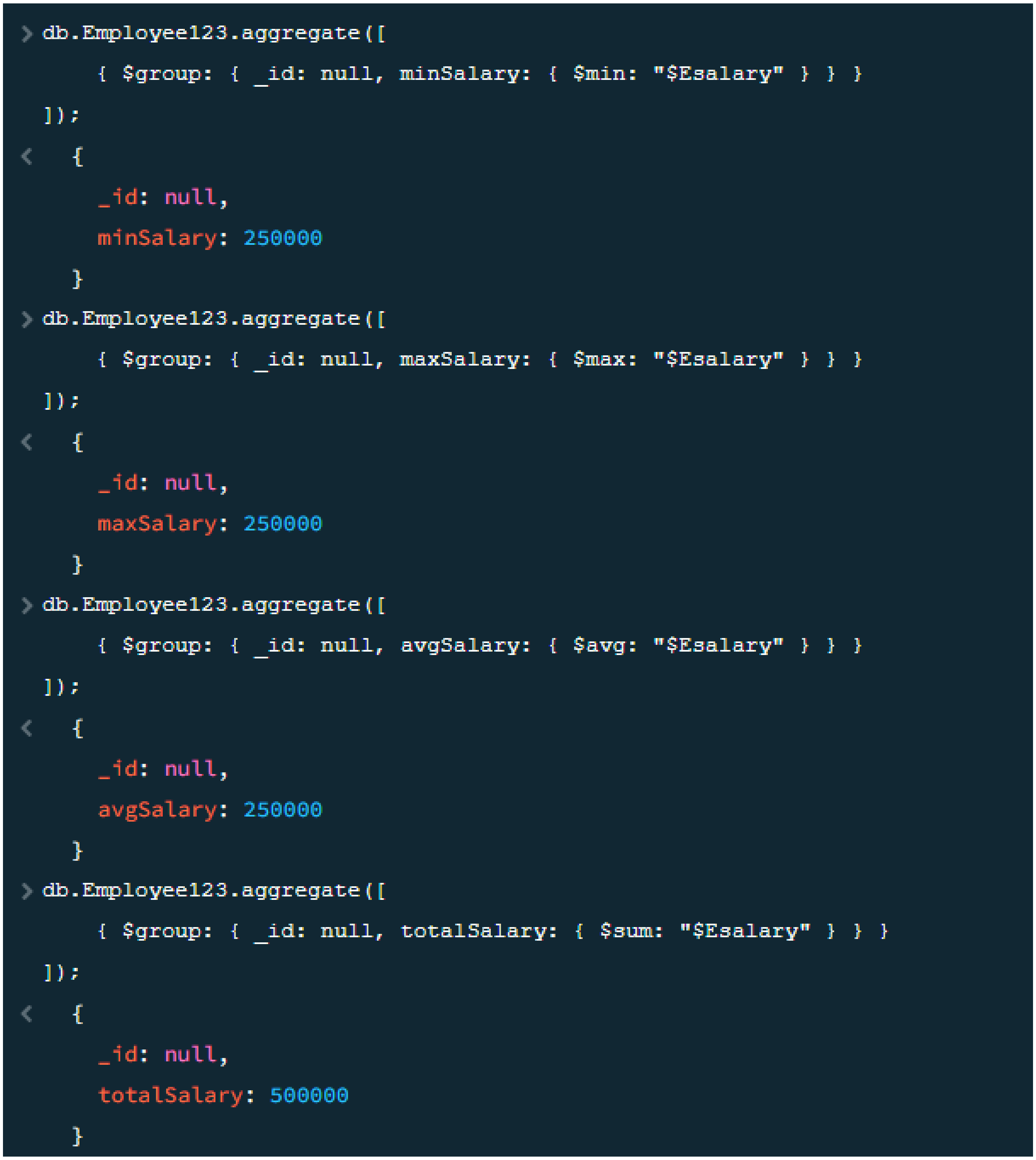
```
> db.Employee123.find({ Esalary: { $lte: 250000 } });
<   {
      _id: ObjectId("66038dcac78b602ec9a44675"),
      Eid: 101,
      Ename: 'Akshay',
      Eaddress: 'Mumbai',
      Emobile: 1234567890,
      Esalary: 250000
    }
    {
      _id: ObjectId("66038dd8c78b602ec9a44676"),
      Eid: 102,
      Ename: 'Arvind',
      Eaddress: 'Mumbai',
      Emobile: 9870654321,
      Esalary: 250000
    }
```

15. Display the details where Eaddress is not Mumbai.



```
> db.Employee123.find({ Eaddress: { $ne: "Mumbai" } });
<   {
      _id: ObjectId("66038e3dc78b602ec9a44677"),
      Eid: 103,
      Ename: 'Sahil',
      Eaddress: 'Pune',
      Emobile: 8765093214
    }
    {
      _id: ObjectId("66038e3dc78b602ec9a44678"),
      Eid: 104,
      Ename: 'Pranay',
      Eaddress: 'Bangalore',
      Emobile: 74310982365
    }
```

16. Perform aggregate functions like min(), max(), avg(), sum().



```
> db.Employee123.aggregate([
      { $group: { _id: null, minSalary: { $min: "$Esalary" } } }
  ]);
<   {
      _id: null,
      minSalary: 250000
    }
> db.Employee123.aggregate([
      { $group: { _id: null, maxSalary: { $max: "$Esalary" } } }
  ]);
<   {
      _id: null,
      maxSalary: 250000
    }
> db.Employee123.aggregate([
      { $group: { _id: null, avgSalary: { $avg: "$Esalary" } } }
  ]);
<   {
      _id: null,
      avgSalary: 250000
    }
> db.Employee123.aggregate([
      { $group: { _id: null, totalSalary: { $sum: "$Esalary" } } }
  ]);
<   {
      _id: null,
      totalSalary: 500000
    }
```