1. Write a query to fetch the EmpFname from the EmployeeInfo collection

db.EmployeeInfo.find({}, {EmpFname : 1})

```
> db.EmployeeInfo.find({}, {EmpFname : 1})
< {
    _id: ObjectId("64cbcdff4f5abd92e5937c43"),
    EmpFname: 'Sanjay'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c44"),
    EmpFname: 'Ananya'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c45"),
    EmpFname: 'Rohan'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c46"),
    EmpFname: 'Sonia'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c47"),
    EmpFname: 'Ankit'
  }
```

2. Write a query to fetch the number of employees working in the department ‘HR’.

db.EmployeeInfo.find({Department : "HR"}).count()

```
> db.EmployeeInfo.find({Department : "HR"}).count()
< 2
```

3. Write a query to get the current date.

db.EmployeeInfo.find({DOB : Date()})

```
> db.EmployeeInfo.find( {DOB : Date()} )
<
```

4. Write a query to retrieve the first four characters of EmpLname from the EmployeeInfo collection.

db.EmployeeInfo.aggregate([{$project: {EmpLNameSubstr: { $substrBytes: ["$EmpLname", 0, 4] }}}])

```
> db.EmployeeInfo.aggregate(
    [
      {
        $project: {
          EmpLNameSubstr: { $substrBytes: [ "$EmpLname", 0, 4 ] }
        }
      }
    ]
  )
< {
    _id: ObjectId("64cbcdff4f5abd92e5937c43"),
    EmpLNameSubstr: 'Mehr'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c44"),
    EmpLNameSubstr: 'Mish'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c45"),
    EmpLNameSubstr: 'Diwa'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c46"),
    EmpLNameSubstr: 'Kulk'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c47"),
    EmpLNameSubstr: 'Kapo'
  }
```

5. Write a query to fetch only the place name(string before brackets) from the Address field of EmployeeInfo collection.

db.EmployeeInfo.aggregate([{$project: {AddressSubstr: { $first: { $split: ["$Address.city", "("] } }}}])

```
> db.EmployeeInfo.aggregate(
    [
      {
        $project: {
          AddressSubstr: {  $first: { $split: [ "$Address.city", "(" ] } }
        }
      }
    ]
  )
< {
    _id: ObjectId("64cbcdff4f5abd92e5937c43"),
    AddressSubstr: 'Hyderabad'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c44"),
    AddressSubstr: 'Delhi'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c45"),
    AddressSubstr: 'Mumbai'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c46"),
    AddressSubstr: 'Hyderabad'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c47"),
    AddressSubstr: 'Delhi'
  }
```

6. Write a query to find all the employees whose salary is between 50000 to 100000.

db.EmployeePosition.find({ $and: [{Salary: {$gt: 50000} }, {Salary: { $lt: 100000 } }] })

```
> db.EmployeePosition.find( { $and: [ {Salary: {$gt: 50000} }, {Salary: { $lt: 100000 } } ] } )
< {
    _id: ObjectId("64cbce3e4f5abd92e5937c4b"),
    EmpID: 2,
    EmpPosition: 'Executive',
    DateOfJoining: '02/05/2022',
    Salary: 75000
  }
  {
    _id: ObjectId("64cbce3e4f5abd92e5937c4c"),
    EmpID: 3,
    EmpPosition: 'Manager',
    DateOfJoining: '01/05/2022',
    Salary: 90000
  }
  {
    _id: ObjectId("64cbce3e4f5abd92e5937c4d"),
    EmpID: 2,
    EmpPosition: 'Lead',
    DateOfJoining: '02/05/2022',
    Salary: 85000
  }
```

7. Write a query to find the names of employees that begin with ‘S’

db.EmployeeInfo.find({EmpFname : /^S/}, {EmpFname:1})

```
> db.EmployeeInfo.find({EmpFname : /^S/}, {EmpFname:1})
< {
    _id: ObjectId("64cbcdff4f5abd92e5937c43"),
    EmpFname: 'Sanjay'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c46"),
    EmpFname: 'Sonia'
  }
```

8. Write a query to retrieve the EmpFname and EmpLname in a single field “FullName”. The first name and the last name must be separated with space.

db.EmployeeInfo.aggregate([{ $project: { fullName: { $concat: ["$EmpFname", " ", "$EmpLname"] } } }])

```
> db.EmployeeInfo.aggregate(
    [
       { $project: { fullName: { $concat: [ "$EmpFname", " ", "$EmpLname" ] } } }
    ]
)
< {
    _id: ObjectId("64cbcdff4f5abd92e5937c43"),
    fullName: 'Sanjay Mehra'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c44"),
    fullName: 'Ananya Mishra'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c45"),
    fullName: 'Rohan Diwan'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c46"),
    fullName: 'Sonia Kulkarni'
  }
  {
    _id: ObjectId("64cbcdff4f5abd92e5937c47"),
    fullName: 'Ankit Kapoor'
  }
```

9. Write a query to fetch all the records from the EmployeeInfo collection ordered by EmpLname in descending order and Department in the ascending order.

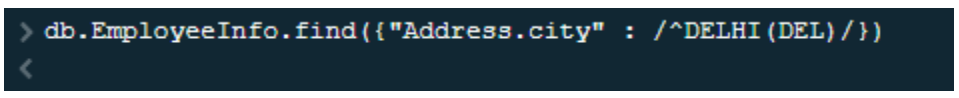
db.EmployeeInfo.find({}).sort({"EmpLname":-1}).sort({"Department":1})

10. Write a query to fetch details of all employees excluding the employees with first names, “Sanjay” and “Sonia” from the EmployeeInfo collection.

db.EmployeeInfo.find({$and: [{ EmpFname : {"$not": /^Sanjay/} }, { EmpFname : {"$not": /^Sonia/} }]})

11. Write a query to fetch details of employees with the address as “DELHI(DEL)”.

db.EmployeeInfo.find({"Address.city" : /^DELHI(DEL)/})



12. Write a query to fetch all employees who also hold the managerial position.

db.EmployeePosition.find({'EmpPosition' : 'Manager'})

```
> db.EmployeePosition.find( {'EmpPosition' : 'Manager'})
< {
    _id: ObjectId("64cbce3e4f5abd92e5937c4a"),
    EmpID: 1,
    EmpPosition: 'Manager',
    DateOfJoining: '01/05/2022',
    Salary: 500000
  }
  {
    _id: ObjectId("64cbce3e4f5abd92e5937c4c"),
    EmpID: 3,
    EmpPosition: 'Manager',
    DateOfJoining: '01/05/2022',
    Salary: 90000
  }
```

13. Write a query to fetch the department-wise count of employees sorted by department's count in ascending order.

db.EmployeePosition.find({}).sort({'EmpID': 1})

```
> db.EmployeePosition.find({}).sort({'EmpID': 1})
< {
    _id: ObjectId("64cbce3e4f5abd92e5937c4a"),
    EmpID: 1,
    EmpPosition: 'Manager',
    DateOfJoining: '01/05/2022',
    Salary: 500000
  }
  {
    _id: ObjectId("64cbce3e4f5abd92e5937c4e"),
    EmpID: 1,
    EmpPosition: 'Executive',
    DateOfJoining: '01/05/2022',
    Salary: 300000
  }
  {
    _id: ObjectId("64cbce3e4f5abd92e5937c4b"),
    EmpID: 2,
    EmpPosition: 'Executive',
    DateOfJoining: '02/05/2022',
    Salary: 75000
  }
  {
    _id: ObjectId("64cbce3e4f5abd92e5937c4d"),
    EmpID: 2,
    EmpPosition: 'Lead',
    DateOfJoining: '02/05/2022',
    Salary: 85000
  }
  {
    _id: ObjectId("64cbce3e4f5abd92e5937c4c"),
    EmpID: 3,
    EmpPosition: 'Manager',
    DateOfJoining: '01/05/2022',
    Salary: 90000
  }
```

14. Write a query to retrieve two minimum and maximum salaries from the EmployeePosition collection.

db.EmployeePosition.find({}).sort({"Salary" : 1}).limit(2)

```
> db.EmployeePosition.find({}).sort({"Salary" : 1}).limit(2)
< {
    _id: ObjectId("64cbce3e4f5abd92e5937c4b"),
    EmpID: 2,
    EmpPosition: 'Executive',
    DateOfJoining: '02/05/2022',
    Salary: 75000
  }
  {
    _id: ObjectId("64cbce3e4f5abd92e5937c4d"),
    EmpID: 2,
    EmpPosition: 'Lead',
    DateOfJoining: '02/05/2022',
    Salary: 85000
  }
```

db.EmployeePosition.find({}).sort({"Salary" : 1}).skip(db.EmployeePosition.countDocuments() - 2)

```
> db.EmployeePosition.find({}).sort({"Salary" : 1}).skip(db.EmployeePosition.countDocuments() - 2)
< {
    _id: ObjectId("64cbce3e4f5abd92e5937c4e"),
    EmpID: 1,
    EmpPosition: 'Executive',
    DateOfJoining: '01/05/2022',
    Salary: 300000
  }
  {
    _id: ObjectId("64cbce3e4f5abd92e5937c4a"),
    EmpID: 1,
    EmpPosition: 'Manager',
    DateOfJoining: '01/05/2022',
    Salary: 500000
  }
```

15. Write a query to retrieve duplicate records from a collection.

db.EmployeeInfo.aggregate([{"$group" : { "_id": "$EmpID", "count": { "$sum": 1 } } },{"$match": {"_id" :{ "$ne" : null } , "count" : {"$gt": 1} } }, {"$project": {"EmpID" : "$_id", "_id" : 0} }]);

```
> db.EmployeeInfo.aggregate([
      {"$group" : { "_id": "$EmpID", "count": { "$sum": 1 } } },
      {"$match": {"_id" :{ "$ne" : null } , "count" : {"$gt": 1} } },
      {"$project": {"EmpID" : "$_id", "_id" : 0} }
  ]);
<
```

16. Write a query to retrieve the list of employees working in the same department.

db.EmployeeInfo.aggregate([{$group: {_id: "$Department",count: { $count: { } }}}])

```
> db.EmployeeInfo.aggregate( [
    {
      $group: {
         _id: "$Department",
         count: { $count: { } }
      }
    }
  ] )
< {
    _id: 'Admin',
    count: 2
  }
  {
    _id: 'Account',
    count: 1
  }
  {
    _id: 'HR',
    count: 2
  }
```

17. Write a query to retrieve the last 3 records from the EmployeeInfo collection.

db.EmployeeInfo.find({}).skip(db.EmployeeInfo.countDocuments() - 3)

```
> db.EmployeeInfo.find({}).skip(db.EmployeeInfo.countDocuments() - 3)
< {
    _id: ObjectId("64cbcdff4f5abd92e5937c45"),
    EmpID: 3,
    EmpFname: 'Rohan',
    EmpLname: 'Diwan',
    Department: 'Account',
    Project: 'P3',
    Address: {
      city: 'Mumbai(BOM)',
      state: 'Maharashtra'
    },
    DOB: {
      month: '1',
      day: '1',
      year: '1980'
    },
    Gender: 'M'
  }
{
    _id: ObjectId("64cbcdff4f5abd92e5937c46"),
    EmpID: 4,
    EmpFname: 'Sonia',
    EmpLname: 'Kulkarni',
    Department: 'HR',
    Project: 'P1',
    Address: {
      city: 'Hyderabad(HYD)',
      state: 'Telangana'
    },
    DOB: {
      month: '2',
      day: '5',
      year: '1992'
    },
    Gender: 'F'
  }
{
    _id: ObjectId("64cbcdff4f5abd92e5937c47"),
    EmpID: 5,
    EmpFname: 'Ankit',
    EmpLname: 'Kapoor',
    Department: 'Admin',
    Project: 'P2',
    Address: {
      city: 'Delhi(DEL)',
      state: ''
    },
    DOB: {
      month: '3',
      day: '7',
      year: '1994'
    },
    Gender: 'M'
  }
```

18. Write a query to find the third-highest salary from the EmpPosition collection.

db.EmployeePosition.find({}).sort({"Salary": -1}).skip(db.EmployeePosition.countDocuments() - 3).limit(1)

```
> db.EmployeePosition.find({}).sort({"Salary": -1}).skip(db.EmployeePosition.countDocuments() - 3).limit(1)
< {
    _id: ObjectId("64cbce3e4f5abd92e5937c4c"),
    EmpID: 3,
    EmpPosition: 'Manager',
    DateOfJoining: '01/05/2022',
    Salary: 90000
  }
```

19. Write a query to display the first and the last record from the EmployeeInfo collection.

```
> db.EmployeeInfo.find({}).limit(1)
< {
    _id: ObjectId("64cbcdff4f5abd92e5937c43"),
    EmpID: 1,
    EmpFname: 'Sanjay',
    EmpLname: 'Mehra',
    Department: 'HR',
    Project: 'P1',
    Address: {
      city: 'Hyderabad(HYD)',
      state: 'Telangana'
    },
    DOB: {
      month: '1',
      day: '12',
      year: '1976'
    },
    Gender: 'M'
  }
```

db.EmployeeInfo.find({}).skip(db.EmployeeInfo.countDocuments() - 1)

```
> db.EmployeeInfo.find({}).skip(db.EmployeeInfo.countDocuments() - 1)
< {
    _id: ObjectId("64cbcdff4f5abd92e5937c47"),
    EmpID: 5,
    EmpFname: 'Ankit',
    EmpLname: 'Kapoor',
    Department: 'Admin',
    Project: 'P2',
    Address: {
      city: 'Delhi(DEL)',
      state: ''
    },
    DOB: {
      month: '3',
      day: '7',
      year: '1994'
    },
    Gender: 'M'
  }
```

20. Write a query to retrieve Departments who have less than 2 employees working in it

db.EmployeeInfo.aggregate([{$group: {_id: "$Department",count: { $sum: 1 }}},{$project: {_id: 0,Department: {$cond: {if: { $lt: ["$count", 2] },then: "$_id",else: "$$REMOVE"}}}}])

```
> db.EmployeeInfo.aggregate([
    {
      $group: {
        _id: "$Department",
        count: { $sum: 1 }
      }
    },
    {
      $project: {
        _id: 0,
        Department: {
          $cond: {
            if: { $lt: ["$count", 2] },
            then: "$_id",
            else: "$$REMOVE"
          }
        }
      }
    }
  ])
< {}
  {
    Department: 'Account'
  }
  {}
```

21. Write a query to retrieve EmpPostion along with total salaries paid for each of them.

db.EmployeePosition.find({}, {EmpPosition : 1, Salary : 1})

```
> db.EmployeePosition.find({}, {EmpPosition : 1, Salary : 1})
< {
    _id: ObjectId("64cbce3e4f5abd92e5937c4a"),
    EmpPosition: 'Manager',
    Salary: 500000
  }
  {
    _id: ObjectId("64cbce3e4f5abd92e5937c4b"),
    EmpPosition: 'Executive',
    Salary: 75000
  }
  {
    _id: ObjectId("64cbce3e4f5abd92e5937c4c"),
    EmpPosition: 'Manager',
    Salary: 90000
  }
  {
    _id: ObjectId("64cbce3e4f5abd92e5937c4d"),
    EmpPosition: 'Lead',
    Salary: 85000
  }
  {
    _id: ObjectId("64cbce3e4f5abd92e5937c4e"),
    EmpPosition: 'Executive',
    Salary: 300000
  }
```