1. Write a MongoDB query to display all the documents in the collection Employee.

```
db.Employee.find()
```

2. Write a MongoDB query to display the fields EmpID, Name, Gender, and salary for all the documents in the collection employee.

```
db.Employee.find( {}, {EmpID:1, EmployeeName:1, Gender:1, Salary:1} )
```

3. Write a MongoDB query to display the fields EID, Name, Gender, and City, but exclude the field _id for all the documents in the collection employee.

```
db.Employee.find( {}, {_id:0, EmpID:1, EmployeeName:1, Gender:1, Salary:1} )
```

4. Write a MongoDB query to display the fields salary, but exclude the field _id for all the documents in the collection employee.

```
db.Employee.find( {}, {_id:false, Salary:true} )
```

5. Write a MongoDB query to display all the Employees which are in the city London.

```
db.Employee.find( {City:"London"} )
```

6. Write a MongoDB query to display the first 5 EID which are in the city Sydney.

```
db.Employee.find( {}, {EmpID:1, City:"Sydney"})
```

7. Write a MongoDB query to display the next 2 Employees after skipping the first 2 which are in the city New York.

```
db.Employee.find( {City:"New York"} ).skip(2).limit(2)
```

8. Write a MongoDB query to display the count of documents in your collection.

```
db.Employee.find().count()
```

9. Write a MongoDB query to display the sum of salary in your collection.

```
db.Employee.aggregate(
                        {$group:
                                {
                                  _id:null,
                                  total:{$sum:"$Salary"}
```

```
                                    }
                                }
                            )


                    -- OR --


        db.Employee.aggregate([
                                {$group:
                                    {
                                        _id:null,
                                        total:{$sum:"$Salary"}
                                    }
                                }
                            ])
```

10.Write a MongoDB query to display the documents whose employee name starts with S or M in your collection.

```
db.Employee.find( {EmployeeName:/^[s,m]/i} )
```

11.Write a MongoDB query to find the employee Id, name, city, and salary for those employees which contain ‘Phi’ as the first three letters of their name.

```
db.Employee.find( {EmployeeName:/^Phi/i}, {EmpID:1, EmployeeName:1, City:1, Salary:1})
```

12.Write a MongoDB query to find the employee Id, name, city, and gender for those employees which contain ‘ael’ as the last three letters of their name.

```
db.Employee.find( {EmployeeName:/ael$/i}, {EmpID:1, EmployeeName:1, City:1, Gender:1})
```

13.Write a MongoDB query to find the name, joining date, and city for those restaurants which contain ‘dne’ as three letters somewhere in their city name.

```
db.Employee.find( {City:/dne/i}, {EmployeeName:1, JoiningDate:1, City:1})
```

14.Write a MongoDB query to find the employee Id, name, city, and joining date for those employees which do not belong to the city London or Sydney.

```
db.Employee.find( {City:{$nin:["London", "Sydney"]}}, {EmpID:1, EmployeeName:1, City:1, JoiningDate:1})
```

15.Write a MongoDB query to find the name and city for those employees which salary

is more than 10000.

```
db.Employee.find( {Salary:{$gt:10000}}, {EmployeeName:1, City:1} )
```

16.Write a MongoDB query to arrange the name of the employees in ascending order along with all the columns.

```
db.Employee.find().sort({EmployeeName:1})
```

17.Write a MongoDB query to arrange the city of the employees in descending order along with all the columns.

```
db.Employee.find().sort({City:-1})
```

18.Write a MongoDB query to arrange the name of the employees in ascending order and, the city should be in descending order.

```
db.Employee.find().sort({EmployeeName:1, City:-1})
```

19.Write a MongoDB query to display city wise sum of salary from employee collection.

```
db.Employee.aggregate(
                        {$group:
                                {
                                  _id:"$City",
                                  total:{$sum:"$Salary"}
                                }
                        }
                      )

            -- OR --

db.Employee.aggregate([
                        {$group:
                                {
                                  _id:"$City",
                                  total:{$sum:"$Salary"}
                                }
                        }
                      ])
```

20.Write a MongoDB query to delete the document whose city name is London.

```
db.Employee.deleteMany({City:"London"})
```