

MongoDB Query Solution

1. Student Database, Create collection, insert data, find, find one, sort, limit, skip, distinct, projection.

Create a student database with the fields: (SRN, Sname, Degree, Sem, CGPA)

Note:

insert 10 documents.

1. Switch to Database
use StudentDB;
2. Create database Collection

```
db.createCollection("Student");
```

3. Insert 10 Records

```
db.Student.insertMany([  
  { SRN: "SRN001", Sname: "Rahul Sharma", Degree:  
  "B.Tech", Sem: 5, CGPA: 8.2 },  
  { SRN: "SRN002", Sname: "Anjali Mehta", Degree: "B.Sc",  
  Sem: 3, CGPA: 7.5 },  
  { SRN: "SRN003", Sname: "Vikram Singh", Degree: "MBA",  
  Sem: 2, CGPA: 8.9 },  
  { SRN: "SRN004", Sname: "Priya Nair", Degree: "BCA",  
  Sem: 4, CGPA: 8.1 },  
  { SRN: "SRN005", Sname: "Rohit Verma", Degree: "BCA",  
  Sem: 6, CGPA: 7.2 },  
  { SRN: "SRN006", Sname: "Sneha Gupta", Degree:  
  "B.Com", Sem: 2, CGPA: 6.8 },  
  { SRN: "SRN007", Sname: "Karan Malhotra", Degree:  
  "B.Tech", Sem: 7, CGPA: 9.1 },  
  { SRN: "SRN008", Sname: "Neha Reddy", Degree: "MBA",  
  Sem: 1, CGPA: 7.9 },  
  { SRN: "SRN009", Sname: "Amit Patel", Degree: "B.Sc",  
  Sem: 5, CGPA: 6.5 },  
])
```

```
    { SRN: "SRN010", Sname: "Divya Rao", Degree: "BCA",  
    Sem: 6, CGPA: 7.8 }  
  );
```

4. Display All Records

```
db.Student.find().pretty();
```

5. Display all the students in BCA

```
db.Student.find({ Degree: "BCA" }).pretty();
```

6. Display all the students in ascending order

```
db.Student.find().sort({ Sname: 1 }).pretty();
```

7. Display first 5 students

```
db.Student.find().limit(5).pretty();
```

8. display students 5,6,7

```
db.Student.find().skip(4).limit(3).pretty();
```

9. list the degree of student "Rahul"

```
db.Student.find(  
  { Sname: "Rahul Sharma" },  
  { Degree: 1, _id: 0 }  
);
```

10. Display students details of 5,6,7 in descending order of percentage

```
db.Student.find()  
  .sort({ CGPA: -1 })  
  .skip(4)  
  .limit(3)  
  .pretty();
```

11. Display the number of students in BCA

```
db.Student.countDocuments({ Degree: "BCA" });
```

12.Display all the degrees without _id

```
db.Student.find({}, { Degree: 1, _id: 0 });
```

13.Display all the distinct degrees

```
db.Student.distinct("Degree");
```

14.Display all the BCA students with CGPA greater than 6, but less than 7.5.

```
db.Student.find(  
  {  
    Degree: "BCA",  
    CGPA: { $gt: 6, $lt: 7.5 }  
  }  
).pretty();
```

15.Display all the students in BCA and in 6th Sem

```
db.Student.find(  
  { Degree: "BCA", Sem: 6 }  
).pretty();
```

Exercise:2

2.Employee Database.Update modifiers (\$set, \$unset, \$inc, \$push, \$pushAll, \$pull, \$pullAll, \$addToSet)

Create an employee database with the fields: {eid, ename, dept, desig, salary, yoj, address{dno,street, locality, city}}

Note:

insert 10 documents.

Questions:

Select/Use a Database

use employeeDB

1. Display all the employees with salary in range (50000, 75000)

```
db.employees.find({  
  salary: { $gte: 50000, $lte: 75000 }  
})
```

2. Display all the employees with designation developer.

```
db.employees.find({  
  desig: "Developer"  
});
```

3. Display the Salary of “Rahul”

```
db.employees.find(  
  { ename: "Rahul" },  
  { _id: 0, ename: 1, salary: 1 }  
);
```

4. Display the city of employee “Rahul”

```
db.employees.find(  
  { ename: "Rahul" },  
  { _id: 0, ename: 1, "address.city": 1 }  
);
```

5. Update the salary of developers by 5000 increment

```
db.employees.updateMany(  
  { desig: "Developer" },  
  { $inc: { salary: 5000 } }  
);
```

6. Add field age to employee “Rahul”

```
db.employees.updateOne(  
  { ename: "Rahul" },  
  { $set: { age: 30 } }  
);
```

7. Remove YOJ from “Rahul”

```
db.employees.updateOne(  
  { ename: "Rahul" },  
  { $unset: { yoj: "" } }  
);
```

8. Add an array field project to “Rahul”

```
db.employees.updateOne(  
  { ename: "Rahul" },  
  { $set: { project: ["Project A", "Project B"] } } // Replace with actual  
projects  
);
```

9. Add p2 and p3 project to “Rahul”

```
db.employees.updateOne(  
  { ename: "Rahul" },  
  { $push: { project: { $each: ["p2", "p3"] } } }  
);
```

10. Remove p3 from “Rahul”

```
db.employees.updateOne(  
  { ename: "Rahul" },  
  { $pull: { project: "p3" } }  
);
```

11. Add a new embedded object “contacts” with “email” and “phone” as array objects to “Rahul”

```
db.employees.updateOne(  
  { ename: "Rahul" },  
  { $set: {  
    contacts: {  
      email: ["rahul@example.com", "r.john@example.com"], // replace  
with actual emails  
      phone: ["123-456-7890", "098-765-4321"] // replace with  
actual phone numbers  
    }  
  }  
  }  
);
```

12. Add two phone numbers to “Rahul”

```
db.employees.updateOne(  
  { ename: "Rahul" },  
  { $push: { "contacts.phone": { $each: ["111-222-3333", "444-555-  
6666"] } } }  
);
```