

1. Find the names and CGPAs of all students from the "IT" department.

Ans: `db.students.find({cgpa:{$gte:8.5}},{name:1,roll_no:1,department:1,_id:0})`

```
>_MONGOSH
> db.students.find({department: "IT"},{name:1,cgpa:1,_id:0})
< {
  name: 'Aditya Patel',
  cgpa: 8.7
}
{
  name: 'Vikas Nair',
  cgpa: 8.4
}
{
  name: 'Karan Gupta',
  cgpa: 8.9
}
{
  name: 'Dev Mishra',
  cgpa: 8.6
}
```

2. List all students whose CGPA is greater than or equal to 8.5. Show only their name, roll number, and department.

Ans: `db.students.find({cgpa:{$gte:8.5}},{name:1,roll_no:1,department:1,_id:0})`

```
>_MONGOSH
> db.students.find({cgpa:{$gte:8.5}},{name:1,roll_no:1,department:1,_id:0})
< {
  roll_no: 'CSE2101',
  name: 'Aarav Sharma',
  department: 'CSE'
}
{
  roll_no: 'CSE2104',
  name: 'Simran Kaur',
  department: 'CSE'
}
{
  roll_no: 'IT2105',
  name: 'Aditya Patel',
  department: 'IT'
}
{
  roll_no: 'CSE2201',
  name: 'Neha Jain',
  department: 'CSE'
}
{
```

3. Find students who are in either the "ECE" or "ME" department using \$in.

Ans: db.students.find({department:{\$in: ['ECE','ME']}})

```
>_MONGOSH
> db.students.find({department:{$in: ['ECE','ME']}})
< {
  _id: ObjectId('684a9548fcc6acc35bc3f4f8'),
  roll_no: 'ECE2102',
  name: 'Priya Verma',
  year: 2,
  department: 'ECE',
  cgpa: 8.2,
  email: 'priya.ece@univ.edu'
}
{
  _id: ObjectId('684a9548fcc6acc35bc3f4f9'),
  roll_no: 'ME2103',
  name: 'Rahul Singh',
  year: 2,
  department: 'ME',
  cgpa: 7.8,
  email: 'rahul.me@univ.edu'
}
{
  _id: ObjectId('684a9548fcc6acc35bc3f4fd'),
  roll_no: 'ECE2202',
```

4. Get the name and CGPA of students who are in the 4th year and have a CGPA greater than 8.0.

Ans: db.students.find({cgpa:{\$gt:8.0},year: 4},{name:1,cgpa:1,_id:0})

```
>_MONGOSH
> db.students.find({cgpa:{$gt:8.0},year: 4},{name:1,cgpa:1,_id:0})
< {
  name: 'Meera Joshi',
  cgpa: 9.3
}
{
  name: 'Ankit Bansal',
  cgpa: 8.1
}
{
  name: 'Karan Gupta',
  cgpa: 8.9
}
```

5. Find all students who are not in the "CSE" department and have a CGPA less than or equal to 7.5 using \$and and \$ne.

Ans: db.students.find({\$and: [{department:{\$ne:'CSE'}},{cgpa:{\$lte:7.5}}]})

```
>_MONGOSH
> db.students.find({$and: [{department:{$ne:'CSE'}},{cgpa:{$lte:7.5}}]})
< {
  _id: ObjectId('684a9548fcc6acc35bc3f502'),
  roll_no: 'ME2303',
  name: 'Pooja Yadav',
  year: 4,
  department: 'ME',
  cgpa: 7.5,
  email: 'pooja.me@univ.edu'
}
```

6. Display the top 3 students with the highest CGPA across all departments.

Ans: db.students.find({}).sort({cgpa:-1}).limit(3)

```
>_MONGOSH
> db.students.find({}).sort({cgpa:-1}).limit(3)
< {
  _id: ObjectId('684a9548fcc6acc35bc3f500'),
  roll_no: 'CSE2301',
  name: 'Meera Joshi',
  year: 4,
  department: 'CSE',
  cgpa: 9.3,
  email: 'meera.cse@univ.edu'
}
{
  _id: ObjectId('684a9548fcc6acc35bc3f4fc'),
  roll_no: 'CSE2201',
  name: 'Neha Jain',
  year: 3,
  department: 'CSE',
  cgpa: 9.1,
  email: 'neha.cse@univ.edu'
}
{
  _id: ObjectId('684a9548fcc6acc35bc3f4fa'),
  roll_no: 'CSE2104',
```

7. Show the next 3 students after skipping the top 3 based on CGPA (pagination).

Ans: `db.students.find({}).sort({cgpa:-1}).limit(3).skip(3)`

```
> db.students.find({}).sort({cgpa:-1}).limit(3).skip(3)
< {
  _id: ObjectId('684a9548fcc6acc35bc3f503'),
  roll_no: 'IT2304',
  name: 'Karan Gupta',
  year: 4,
  department: 'IT',
  cgpa: 8.9,
  email: 'karan.it@univ.edu'
}
{
  _id: ObjectId('684a9548fcc6acc35bc3f508'),
  roll_no: 'CSE2405',
  name: 'Tanya Roy',
  year: 1,
  department: 'CSE',
  cgpa: 8.7,
  email: 'tanya.cse@univ.edu'
}
{
  _id: ObjectId('684a9548fcc6acc35bc3f4fb'),
  roll_no: 'IT2105',
```

8. Increase the CGPA by 0.2 for all students in the "CSE" department.

Ans: `db.students.updateMany({department:"CSE"},{$inc:{cgpa:0.2}})`

```
> _MONGOSH
```

```
> db.students.updateMany({department:"CSE"},{$inc:{cgpa:0.2}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 6,
  modifiedCount: 6,
  upsertedCount: 0
}
```

9. Set the department to "Probation" for all students with CGPA less than 7.0.

Ans: `db.students.updateMany({cgpa:{$lt:7.0}},{$set: {department:'Probation'}})`

>_MONGOSH

```
> db.students.updateMany({cgpa:{$lt:7.0}},{$set: {department:'Probation'}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

10. Remove the email field from all students in the "ME" department.

Ans: `db.students.updateMany({department:"ME"},{$unset:{email:""}})`

>_MONGOSH

```
> db.students.updateMany({department:"ME"},{$unset:{email:""}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 5,
  modifiedCount: 0,
  upsertedCount: 0
}
```

11. Rename the field cgpa to gpa for all students.

Ans: `db.students.updateMany({},{$rename:{cgpa: 'gpa'}})`

>_MONGOSH

```
> db.students.updateMany({},{$rename:{cgpa: 'gpa'}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 20,
  modifiedCount: 0,
  upsertedCount: 0
}
```

12. Replace the entire document of a student with roll number "CSE2104" with a new record containing updated values.

Ans: `db.students.updateOne({roll_no: "CSE2104"},{$set:{ roll_no: "CSE2104", name: "Prabhav Maurya", year: 1, department: "ECE", cgpa: 9.1, email: "prabhav.ece@univ.edu" }})`

```
>_MONGOSH
> db.students.updateOne({roll_no: "CSE2104"},{$set:{ roll_no: "CSE2104", name: "Prabhav Maurya", year: 1, department: "ECE", cgpa: 9.1, email: "prabhav.ece@univ.edu" }})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Before update

After Update

```
> db.students.find({roll_no:"CSE2104"})
< {
  _id: ObjectId('684a9548fcc6acc35bc3f4fa'),
  roll_no: 'CSE2104',
  name: 'Simran Kaur',
  year: 2,
  department: 'CSE',
  email: 'simran.cse@univ.edu',
  gpa: 9.2,
  cgpa: 9
}
```

```
> db.students.find({roll_no:"CSE2104"})
< {
  _id: ObjectId('684a9548fcc6acc35bc3f4fa'),
  roll_no: 'CSE2104',
  name: 'Prabhav Maurya',
  year: 1,
  department: 'ECE',
  email: 'prabhav.ece@univ.edu',
  gpa: 9,
  cgpa: 9.1
}
```