

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

```
db.students.find({ department:"IT" }, {name:1, cgpa:1, _id: 0 })
```

```
> 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.

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

```
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'
}
{
  roll_no: 'CSE2301',
  name: 'Meera Joshi',
  department: 'CSE'
}
{
  roll_no: 'IT2304',
  name: 'Karan Gupta',
  department: 'IT'
}
{
  roll_no: 'IT2404',
  name: 'Dev Mishra',
```

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

```
db.students.find({ department: { $in: ["ECE", "ME"] } })
```

```

> db.students.find({ department: { $in: ["ECE", "ME"] } })
< {
  _id: ObjectId('684e9e01a13f679364091e9b'),
  roll_no: 'ECE2102',
  name: 'Priya Verma',
  year: 2,
  department: 'ECE',
  cgpa: 8.2,
  email: 'priya.ece@univ.edu'
}
{
  _id: ObjectId('684e9e01a13f679364091e9c'),
  roll_no: 'ME2103',
  name: 'Rahul Singh',
  year: 2,
  department: 'ME',
  cgpa: 7.8,
  email: 'rahul.me@univ.edu'
}
{
  _id: ObjectId('684e9e01a13f679364091ea0'),
  roll_no: 'ECE2202',
  name: 'Rohan Mehta',
  year: 3,
  department: 'ECE',
  cgpa: 8.3,
  email: 'rohan.ece@univ.edu'
}
{
  _id: ObjectId('684e9e01a13f679364091ea1'),
  roll_no: 'ME2203',
  name: 'Sneha Reddy',
  year: 3,
  department: 'ME',

```

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

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

```
> db.students.find({ year: 4, cgpa: { $gt: 8.0 } }, { 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.

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

```
> db.students.find({ $and: [ { department: { $ne: "CSE" } }, { cgpa: { $lte: 7.5 } } ] })
< {
  _id: ObjectId('684e9e01a13f679364091ea5'),
  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.

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

```
> db.students.find().sort({ cgpa: -1 }).limit(3)
< {
  _id: ObjectId('684e9e01a13f679364091ea3'),
  roll_no: 'CSE2301',
  name: 'Meera Joshi',
  year: 4,
  department: 'CSE',
  cgpa: 9.3,
  email: 'meera.cse@univ.edu'
}
{
  _id: ObjectId('684e9e01a13f679364091e9f'),
  roll_no: 'CSE2201',
  name: 'Neha Jain',
  year: 3,
  department: 'CSE',
  cgpa: 9.1,
  email: 'neha.cse@univ.edu'
}
{
  _id: ObjectId('684e9e01a13f679364091e9d'),
  roll_no: 'CSE2104',
  name: 'Simran Kaur',
  year: 2,
  department: 'CSE',
  cgpa: 9,
  email: 'simran.cse@univ.edu'
}
```

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

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

```
> db.students.find().sort({ cgpa: -1 }).skip(3).limit(3)
< {
  _id: ObjectId('684e9e01a13f679364091ea6'),
  roll_no: 'IT2304',
  name: 'Karan Gupta',
  year: 4,
  department: 'IT',
  cgpa: 8.9,
  email: 'karan.it@univ.edu'
}
{
  _id: ObjectId('684e9e01a13f679364091eab'),
  roll_no: 'CSE2405',
  name: 'Tanya Roy',
  year: 1,
  department: 'CSE',
  cgpa: 8.7,
  email: 'tanya.cse@univ.edu'
}
{
  _id: ObjectId('684e9e01a13f679364091e9e'),
  roll_no: 'IT2105',
  name: 'Aditya Patel',
  year: 2,
  department: 'IT',
  cgpa: 8.7,
  email: 'aditya.it@univ.edu'
}
```

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

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

```
> 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.

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

```
> 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.

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

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

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

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

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

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