

## Assignment-1 (Mongodb Queries)

### Find / Query Operations

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

```
a1> db.students.find({department:"IT"},{_id:0,name:1,cgpa:1})
[
  { 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.

```
a1> db.students.find({cgpa:{$gte:8.5}},{_id:0,name:1,roll_no:1,department:1})
[
  { 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', department: 'IT' },
  { roll_no: 'CSE2405', name: 'Tanya Roy', department: 'CSE' }
]
```

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

```
a1> db.students.find(
... {department:{$in:["ECE","ME"]}},
... {_id:0,roll_no:1,name:1}
... )
[
  { roll_no: 'ECE2102', name: 'Priya Verma' },
  { roll_no: 'ME2103', name: 'Rahul Singh' },
  { roll_no: 'ECE2202', name: 'Rohan Mehta' },
  { roll_no: 'ME2203', name: 'Sneha Reddy' },
  { roll_no: 'ECE2302', name: 'Ankit Bansal' },
  { roll_no: 'ME2303', name: 'Pooja Yadav' },
  { roll_no: 'ECE2402', name: 'Aman Khan' },
  { roll_no: 'ME2403', name: 'Ishita Desai' },
  { roll_no: 'ECE2406', name: 'Nikhil Rao' },
  { roll_no: 'ME2407', name: 'Shreya Pandey' }
]
```

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

```
a1> db.students.find(
... {year:4,cgpa:{$gt:8.0}},
... {_id:0,name:1,cgpa:1}
... )
[
  { 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.

```
a1> db.students.find({$and:[{department:{$ne:"CSE"}},{cgpa:{$lte:7.5}}]})
[
  {
    _id: ObjectId('684d6c45e09089a86950eb72'),
    roll_no: 'ME2303',
    name: 'Pooja Yadav',
    year: 4,
    department: 'ME',
    cgpa: 7.5,
    email: 'pooja.me@univ.edu'
  }
]
```

## Sort / Pagination

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

```
a1> db.students.find().sort({cgpa:-1}).limit(3)
[
  {
    _id: ObjectId('684d6c45e09089a86950eb70'),
    roll_no: 'CSE2301',
    name: 'Meera Joshi',
    year: 4,
    department: 'CSE',
    cgpa: 9.3,
    email: 'meera.cse@univ.edu'
  },
  {
    _id: ObjectId('684d6c45e09089a86950eb6c'),
    roll_no: 'CSE2201',
    name: 'Neha Jain',
    year: 3,
    department: 'CSE',
    cgpa: 9.1,
    email: 'neha.cse@univ.edu'
  },
  {
    _id: ObjectId('684d6c45e09089a86950eb6a'),
    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).

```

a1> db.students.find().sort({cgpa:-1}).skip(3).limit(3)
[
  {
    _id: ObjectId('684d6c45e09089a86950eb73'),
    roll_no: 'IT2304',
    name: 'Karan Gupta',
    year: 4,
    department: 'IT',
    cgpa: 8.9,
    email: 'karan.it@univ.edu'
  },
  {
    _id: ObjectId('684d6c45e09089a86950eb78'),
    roll_no: 'CSE2405',
    name: 'Tanya Roy',
    year: 1,
    department: 'CSE',
    cgpa: 8.7,
    email: 'tanya.cse@univ.edu'
  },
  {
    _id: ObjectId('684d6c45e09089a86950eb6b'),
    roll_no: 'IT2105',
    name: 'Aditya Patel',
    year: 2,
    department: 'IT',
    cgpa: 8.7,
    email: 'aditya.it@univ.edu'
  }
]

```

## Update Operations

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

```

a1> 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.

```

a1> 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.

```

a1> db.students.updateMany(
... {department:"ME"},
... {$unset:{email:1}}
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 5,
  modifiedCount: 5,
  upsertedCount: 0
}

```

## Bonus (Advanced)

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

```
a1> 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.

```
a1> db.students.replaceOne(
... {roll_no: "CSE2104"},
... {roll_no: "CSE2104", name: "Simran Kaur", year: 2, department: "CSE", cgpa: 9.5, email: "simrankaur.cse@univ.edu"}
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```