BABU BANARASI DAS UNIVERSITY

SESSION:2025-26

Name: Naincy Saroj

Roll NO: 58

Program: BCA (DS&AI)

Batch: BCADS24

Semester: 3rd

Subject Teacher: Ankit Verma

Signature: -

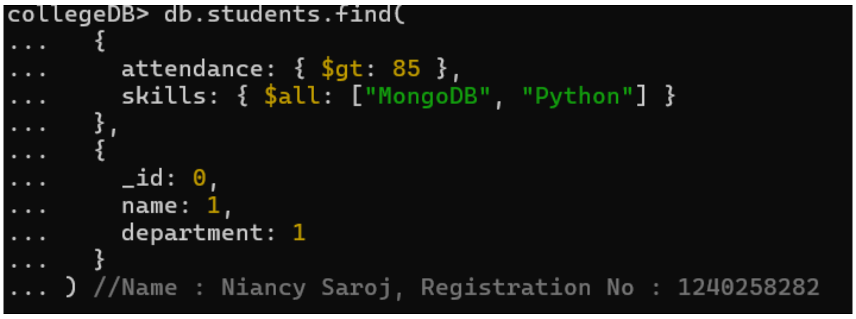
INDEX

ASSESSMENT TASK

1.Complex Filters & Projections

Q1. List the names and departments of students who have more than 85% attendance and are skilled in both "MongoDB" and "Python".

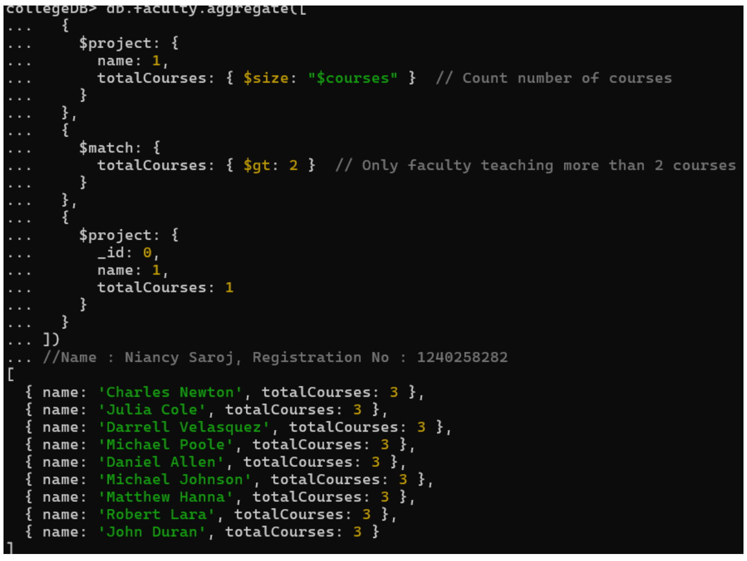
Output:



```
collegeDB> db.students.find(
...    {
...      attendance: { $gt: 85 },
...      skills: { $all: ["MongoDB", "Python"] }
...    },
...    {
...      _id: 0,
...      name: 1,
...      department: 1
...    }
... ) //Name : Niancy Saroj, Registration No : 1240258282
```

- attendance: { $gt: 85 } → selects students with attendance greater than 85%.
- skills: {$all: ["MongoDB", "Python"] } → selects students who have both "MongoDB" and "Python" in their skills array.
- In the projection part, specify which fields to display:
- { name: 1, department: 1, _id: 0 } → shows only name and department, hides the _id field.

Q2. Show all faculty who are teaching more than 2 courses. Display their names and the total number of courses they teach.

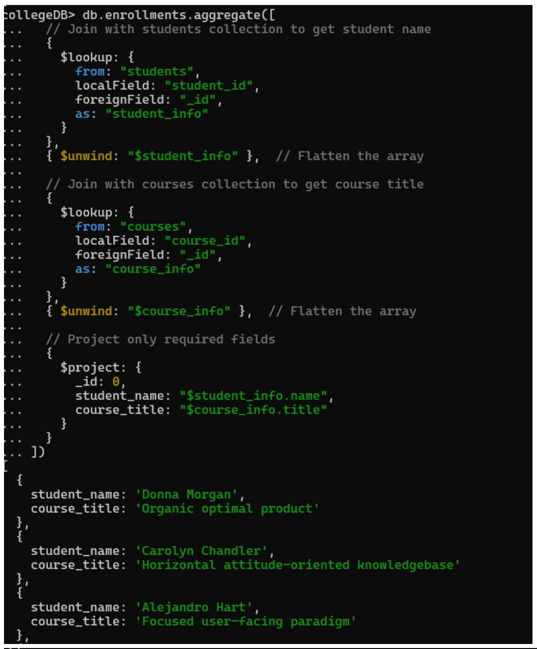


```
collegeDB> db.faculty.aggregate([
...    {
...      $project: {
...        name: 1,
...        totalCourses: { $size: "$courses" }  // Count number of courses
...      }
...    },
...    {
...      $match: {
...        totalCourses: { $gt: 2 }  // Only faculty teaching more than 2 courses
...      }
...    },
...    {
...      $project: {
...        _id: 0,
...        name: 1,
...        totalCourses: 1
...      }
...    }
... ])
... //Name : Niancy Saroj, Registration No : 1240258282
[
  { name: 'Charles Newton', totalCourses: 3 },
  { name: 'Julia Cole', totalCourses: 3 },
  { name: 'Darrell Velasquez', totalCourses: 3 },
  { name: 'Michael Poole', totalCourses: 3 },
  { name: 'Daniel Allen', totalCourses: 3 },
  { name: 'Michael Johnson', totalCourses: 3 },
  { name: 'Matthew Hanna', totalCourses: 3 },
  { name: 'Robert Lara', totalCourses: 3 },
  { name: 'John Duran', totalCourses: 3 }
]
```

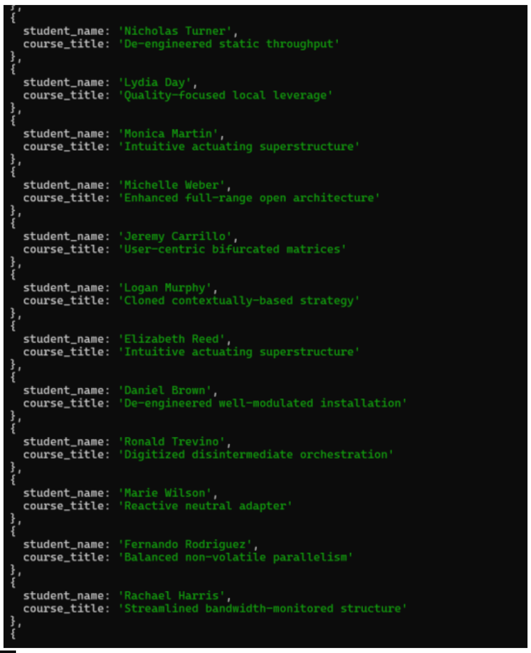
- $size: "$courses" → counts how many courses each faculty teaches
- Use $match to filter faculty members who are teaching more than 2 courses, using total Courses: { $gt: 2 }.
- Use $project to display only the fields needed in output: name and totalCourses, hide _id.

2. Joins ($lookup) and Aggregations

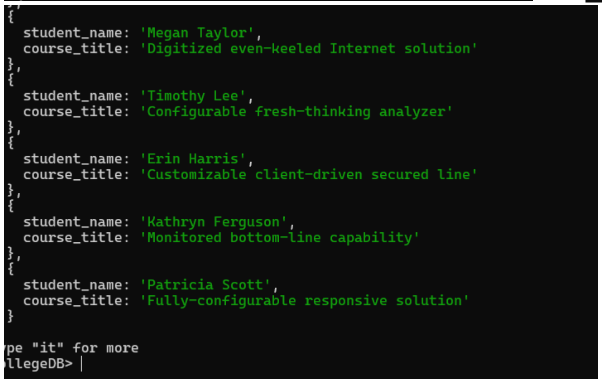
Q3. Write a query to show each student's name along with the course titles they are enrolled in (use $lookup between enrollments, students, and courses).



```
ollegeDB> db.enrollments.aggregate([
...   // Join with students collection to get student name
...   {
...     $lookup: {
...       from: "students",
...       localField: "student_id",
...       foreignField: "_id",
...       as: "student_info"
...     }
...   },
...   { $unwind: "$student_info" },  // Flatten the array
...
...   // Join with courses collection to get course title
...   {
...     $lookup: {
...       from: "courses",
...       localField: "course_id",
...       foreignField: "_id",
...       as: "course_info"
...     }
...   },
...   { $unwind: "$course_info" },  // Flatten the array
...
...   // Project only required fields
...   {
...     $project: {
...       _id: 0,
...       student_name: "$student_info.name",
...       course_title: "$course_info.title"
...     }
...   }
... ])
[
  {
    student_name: 'Donna Morgan',
    course_title: 'Organic optimal product'
  },
  {
    student_name: 'Carolyn Chandler',
    course_title: 'Horizontal attitude-oriented knowledgebase'
  },
  {
    student_name: 'Alejandro Hart',
    course_title: 'Focused user-facing paradigm'
  },
```



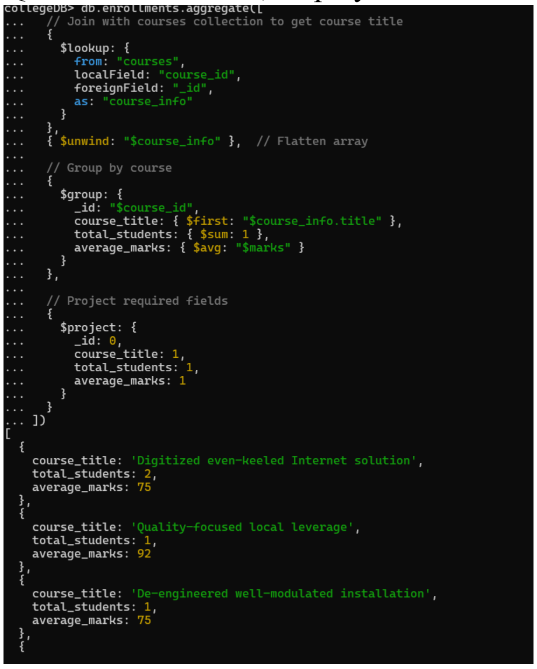
```
},
{
  student_name: 'Nicholas Turner',
  course_title: 'De-engineered static throughput'
},
{
  student_name: 'Lydia Day',
  course_title: 'Quality-focused local leverage'
},
{
  student_name: 'Monica Martin',
  course_title: 'Intuitive actuating superstructure'
},
{
  student_name: 'Michelle Weber',
  course_title: 'Enhanced full-range open architecture'
},
{
  student_name: 'Jeremy Carrillo',
  course_title: 'User-centric bifurcated matrices'
},
{
  student_name: 'Logan Murphy',
  course_title: 'Cloned contextually-based strategy'
},
{
  student_name: 'Elizabeth Reed',
  course_title: 'Intuitive actuating superstructure'
},
{
  student_name: 'Daniel Brown',
  course_title: 'De-engineered well-modulated installation'
},
{
  student_name: 'Ronald Trevino',
  course_title: 'Digitized disintermediate orchestration'
},
{
  student_name: 'Marie Wilson',
  course_title: 'Reactive neutral adapter'
},
{
  student_name: 'Fernando Rodriguez',
  course_title: 'Balanced non-volatile parallelism'
},
{
  student_name: 'Rachael Harris',
  course_title: 'Streamlined bandwidth-monitored structure'
},
{
```



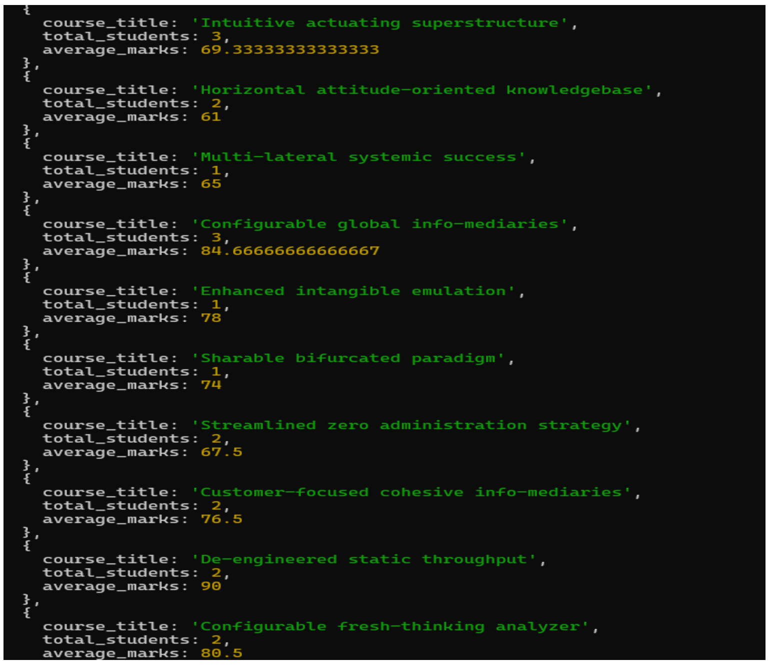
```
},
{
  student_name: 'Megan Taylor',
  course_title: 'Digitized even-keeled Internet solution'
},
{
  student_name: 'Timothy Lee',
  course_title: 'Configurable fresh-thinking analyzer'
},
{
  student_name: 'Erin Harris',
  course_title: 'Customizable client-driven secured line'
},
{
  student_name: 'Kathryn Ferguson',
  course_title: 'Monitored bottom-line capability'
},
{
  student_name: 'Patricia Scott',
  course_title: 'Fully-configurable responsive solution'
}

pe "it" for more
llegeDB> |
```

- Use $project to show only what we need:
- Student's name → from student_info.name
- Course title → from course_info.title
- Hide _id so output looks clean.

Q4. For each course, display the course title, number of students enrolled, and average marks (use $group)



```
collegeDB> db.enrollments.aggregate([
...   // Join with courses collection to get course title
...   {
...     $lookup: {
...       from: "courses",
...       localField: "course_id",
...       foreignField: "_id",
...       as: "course_info"
...     }
...   },
...   { $unwind: "$course_info" },  // Flatten array
...
...   // Group by course
...   {
...     $group: {
...       _id: "$course_id",
...       course_title: { $first: "$course_info.title" },
...       total_students: { $sum: 1 },
...       average_marks: { $avg: "$marks" }
...     }
...   },
...
...   // Project required fields
...   {
...     $project: {
...       _id: 0,
...       course_title: 1,
...       total_students: 1,
...       average_marks: 1
...     }
...   }
... ])
[
  {
    course_title: 'Digitized even-keeled Internet solution',
    total_students: 2,
    average_marks: 75
  },
  {
    course_title: 'Quality-focused local leverage',
    total_students: 1,
    average_marks: 92
  },
  {
    course_title: 'De-engineered well-modulated installation',
    total_students: 1,
    average_marks: 75
  },
  {
```

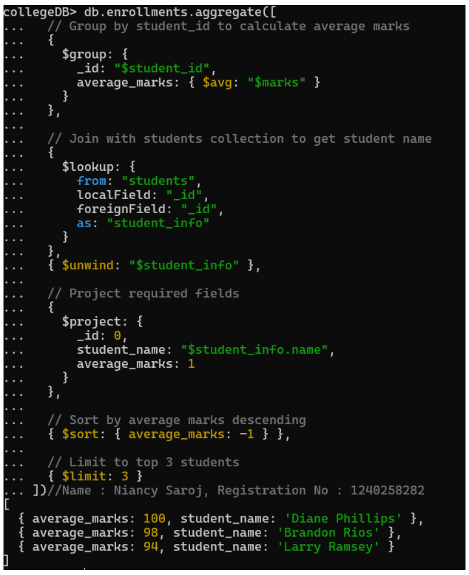


```
  {
    course_title: 'Intuitive actuating superstructure',
    total_students: 3,
    average_marks: 69.33333333333333
  },
  {
    course_title: 'Horizontal attitude-oriented knowledgebase',
    total_students: 2,
    average_marks: 61
  },
  {
    course_title: 'Multi-lateral systemic success',
    total_students: 1,
    average_marks: 65
  },
  {
    course_title: 'Configurable global info-mediaries',
    total_students: 3,
    average_marks: 84.66666666666667
  },
  {
    course_title: 'Enhanced intangible emulation',
    total_students: 1,
    average_marks: 78
  },
  {
    course_title: 'Sharable bifurcated paradigm',
    total_students: 1,
    average_marks: 74
  },
  {
    course_title: 'Streamlined zero administration strategy',
    total_students: 2,
    average_marks: 67.5
  },
  {
    course_title: 'Customer-focused cohesive info-mediaries',
    total_students: 2,
    average_marks: 76.5
  },
  {
    course_title: 'De-engineered static throughput',
    total_students: 2,
    average_marks: 90
  },
  {
    course_title: 'Configurable fresh-thinking analyzer',
    total_students: 2,
    average_marks: 80.5
```

- total _students → counts how many students are enrolled using $sum: 1
- avg_marks → finds average marks using $avg: "$marks".
- Use $lookup to join this data with the courses collection.

3.Grouping, Sorting, and Limiting

Q5. Find the top 3 students with the highest average marks across all enrolled courses



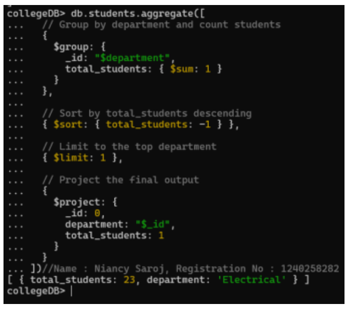
```
collegeDB> db.enrollments.aggregate([
...   // Group by student_id to calculate average marks
...   {
...     $group: {
...       _id: "$student_id",
...       average_marks: { $avg: "$marks" }
...     }
...   },
...
...   // Join with students collection to get student name
...   {
...     $lookup: {
...       from: "students",
...       localField: "_id",
...       foreignField: "_id",
...       as: "student_info"
...     }
...   },
...   { $unwind: "$student_info" },
...
...   // Project required fields
...   {
...     $project: {
...       _id: 0,
...       student_name: "$student_info.name",
...       average_marks: 1
...     }
...   },
...
...   // Sort by average marks descending
...   { $sort: { average_marks: -1 } },
...
...   // Limit to top 3 students
...   { $limit: 3 }
... ])//Name : Niancy Saroj, Registration No : 1240258282
[
  { average_marks: 100, student_name: 'Diane Phillips' },
  { average_marks: 98, student_name: 'Brandon Rios' },
  { average_marks: 94, student_name: 'Larry Ramsey' }
]
```

- Use $group to collect marks of each student by student_id.
- Use $avg: "$marks" to calculate average marks for each student.
- Use $lookup to join with the students collection.

Q6. Count how many students are in each department. Display the department with the highest number of students.

Output:

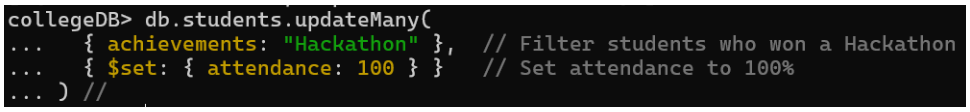
- total_students: { $sum: 1 } → counts how many students are there in each department
- Use $sort: { total_students: -1 } to arrange the results in descending order (highest first).



```
collegeDB> db.students.aggregate([
...   // Group by department and count students
...   {
...     $group: {
...       _id: "$department",
...       total_students: { $sum: 1 }
...     }
...   },
...
...   // Sort by total_students descending
...   { $sort: { total_students: -1 } },
...
...   // Limit to the top department
...   { $limit: 1 },
...
...   // Project the final output
...   {
...     $project: {
...       _id: 0,
...       department: "$_id",
...       total_students: 1
...     }
...   }
... ])//Name : Niancy Saroj, Registration No : 1240258282
[ { total_students: 23, department: 'Electrical' } ]
collegeDB> |
```

4.Update, Upsert, and Delete

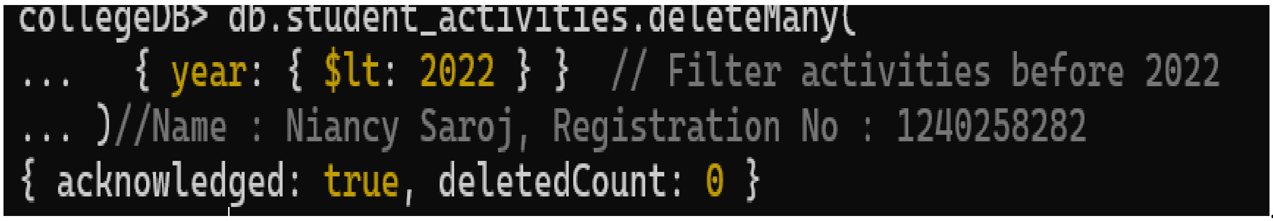
Q7. Update attendance to 100% for all students who won any "Hackathon".



```
collegeDB> db.students.updateMany(
...    { achievements: "Hackathon" },  // Filter students who won a Hackathon
...    { $set: { attendance: 100 } }   // Set attendance to 100%
... ) //
```

- The filter condition is { "activities.name": "Hackathon" }
- { $set: { attendance: 100 } } → sets attendance to 100%.

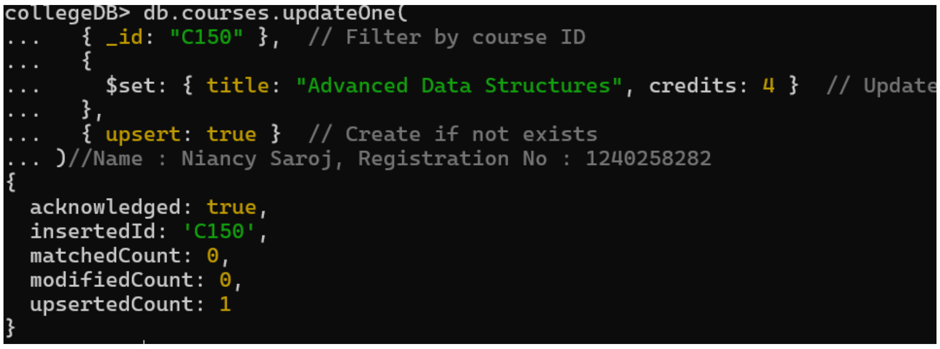
Q8. Delete all student activity records where the activity year is before 2022



```
collegeDB> db.student_activities.deleteMany(
...    { year: { $lt: 2022 } }  // Filter activities before 2022
... )//Name : Niancy Saroj, Registration No : 1240258282
{ acknowledged: true, deletedCount: 0 }
```

- The filter condition is { year: { $lt: 2022 } }
- This selects all activity records where the year field is less than 20
- Run the query - all old activity records will be removed.

Q9. Upsert a course record for "Data Structures" with ID "C150" and credits 4—if it doesn't exist, insert it; otherwise update its title to "Advanced Data Structures".



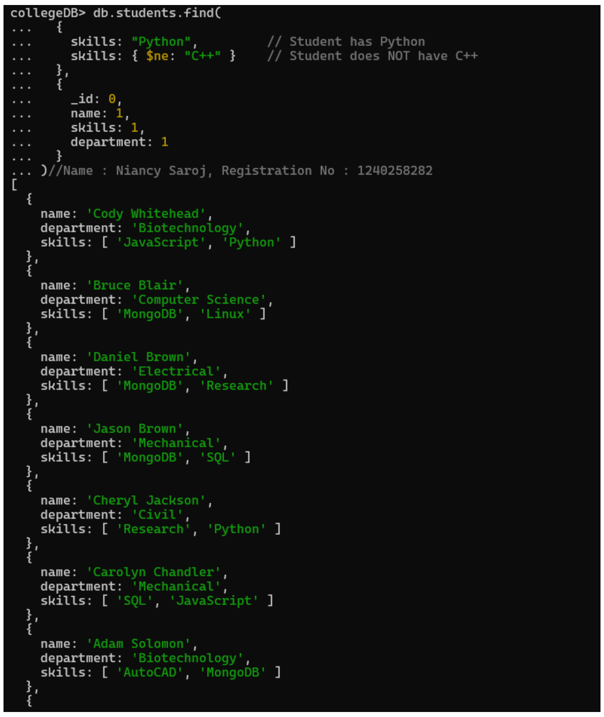
```
collegeDB> db.courses.updateOne(
...    { _id: "C150" },  // Filter by course ID
...    {
...      $set: { title: "Advanced Data Structures", credits: 4 }  // Update
...    },
...    { upsert: true }  // Create if not exists
... )//Name : Niancy Saroj, Registration No : 1240258282
{
  acknowledged: true,
  insertedId: 'C150',
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
```

- Filter: { _id: "C150" } → finds the course with ID "C150".
- Update Part: { $set: { title: "Advanced Data Structures", credits: 4 } }
- Upsert Option: { upsert: true }

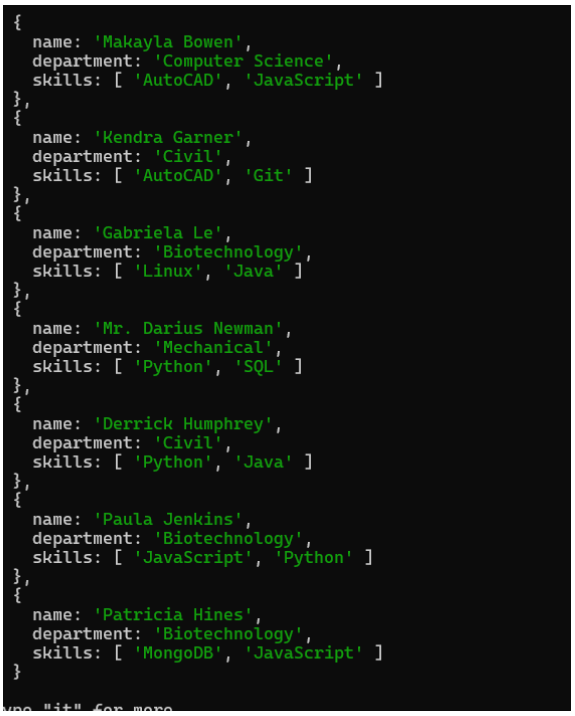
5. Array & Operator Usage

Q10. Find all students who have "Python" as a skill but not "C++".

- skills: "Python" → selects students who have "Python" in their skills array.
- skills: { $ne: "C++" } → ensures the student does not have "C++" in their skills array.

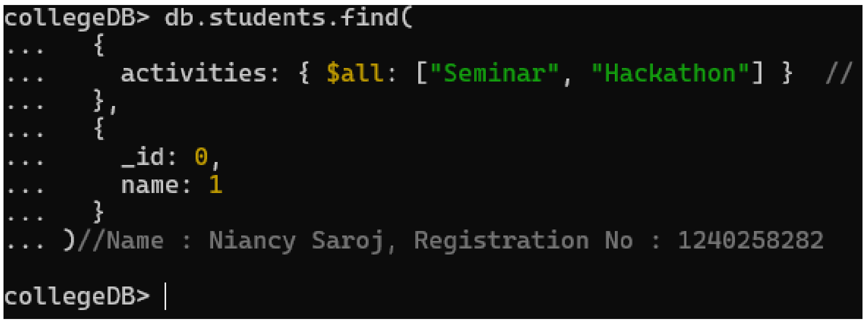


```
collegeDB> db.students.find(
...   {
...     skills: "Python",          // Student has Python
...     skills: { $ne: "C++" }    // Student does NOT have C++
...   },
...   {
...     _id: 0,
...     name: 1,
...     skills: 1,
...     department: 1
...   }
... )//Name : Niancy Saroj, Registration No : 1240258282
[
  {
    name: 'Cody Whitehead',
    department: 'Biotechnology',
    skills: [ 'JavaScript', 'Python' ]
  },
  {
    name: 'Bruce Blair',
    department: 'Computer Science',
    skills: [ 'MongoDB', 'Linux' ]
  },
  {
    name: 'Daniel Brown',
    department: 'Electrical',
    skills: [ 'MongoDB', 'Research' ]
  },
  {
    name: 'Jason Brown',
    department: 'Mechanical',
    skills: [ 'MongoDB', 'SQL' ]
  },
  {
    name: 'Cheryl Jackson',
    department: 'Civil',
    skills: [ 'Research', 'Python' ]
  },
  {
    name: 'Carolyn Chandler',
    department: 'Mechanical',
    skills: [ 'SQL', 'JavaScript' ]
  },
  {
    name: 'Adam Solomon',
    department: 'Biotechnology',
    skills: [ 'AutoCAD', 'MongoDB' ]
  },
  {
```



```
{
  name: 'Makayla Bowen',
  department: 'Computer Science',
  skills: [ 'AutoCAD', 'JavaScript' ]
},
{
  name: 'Kendra Garner',
  department: 'Civil',
  skills: [ 'AutoCAD', 'Git' ]
},
{
  name: 'Gabriela Le',
  department: 'Biotechnology',
  skills: [ 'Linux', 'Java' ]
},
{
  name: 'Mr. Darius Newman',
  department: 'Mechanical',
  skills: [ 'Python', 'SQL' ]
},
{
  name: 'Derrick Humphrey',
  department: 'Civil',
  skills: [ 'Python', 'Java' ]
},
{
  name: 'Paula Jenkins',
  department: 'Biotechnology',
  skills: [ 'JavaScript', 'Python' ]
},
{
  name: 'Patricia Hines',
  department: 'Biotechnology',
  skills: [ 'MongoDB', 'JavaScript' ]
}
ype "it" for more
```

Q11. Return names of students who participated in "Seminar" and "Hackathon" both.

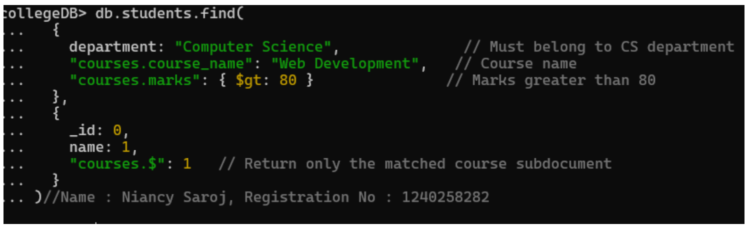


```
collegeDB> db.students.find(
...   {
...     activities: { $all: ["Seminar", "Hackathon"] }  //
...   },
...   {
...     _id: 0,
...     name: 1
...   }
... )//Name : Niancy Saroj, Registration No : 1240258282

collegeDB> |
```

- Condition: activities: { $all: ["Seminar", "Hackathon"] }
- $all ensures the student has both "Seminar" and "Hackathon" in their activities array.
- Projection: { _id: 0, name: 1 }

6.Subdocuments and Nested Conditions

Q12. Find students who scored more than 80 in "Web Development" only if they belong to the "Computer Science" department.



```
ollegeDB> db.students.find(
..   {
..     department: "Computer Science",               // Must belong to CS department
..     "courses.course_name": "Web Development",   // Course name
..     "courses.marks": { $gt: 80 }                // Marks greater than 80
..   },
..   {
..     _id: 0,
..     name: 1,
..     "courses.$": 1   // Return only the matched course subdocument
..   }
.. )//Name : Niancy Saroj, Registration No : 1240258282
```

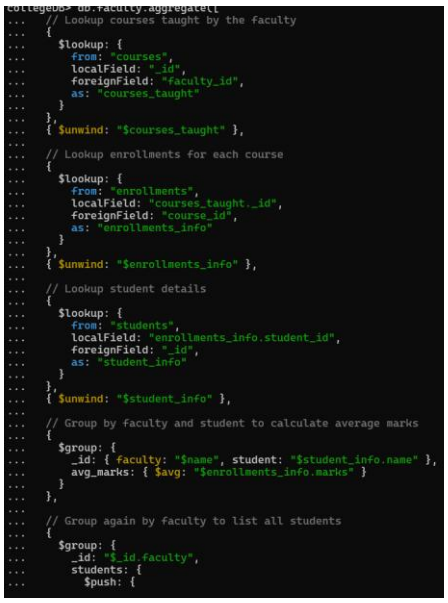
"courses.course_name": "Web Development" → ensures the student has that course.

"courses.marks": { $gt: 80 } → ensures marks in that course are greater than 80.

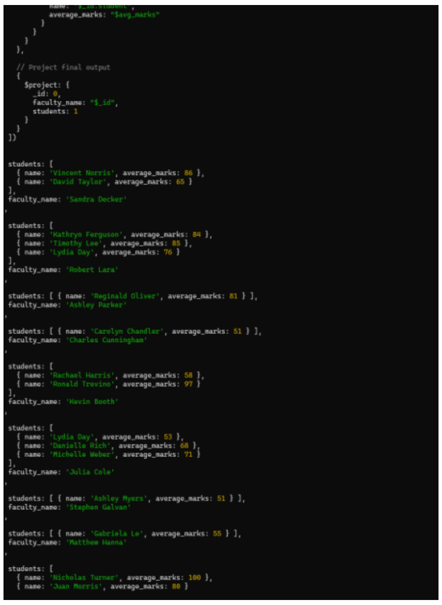
Projection: { _id: 0, name: 1, "course

7. Advanced Aggregation (Challenge Level)

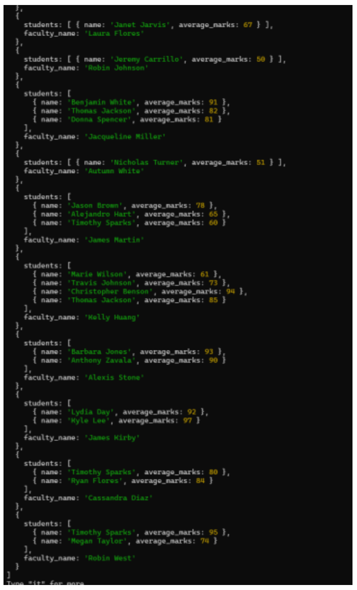
Q13. For each faculty member, list the names of all students enrolled in their courses along with average marks per student per faculty.



```
collegeDB> db.faculty.aggregate([
...   // Lookup courses taught by the faculty
...   {
...     $lookup: {
...       from: "courses",
...       localField: "_id",
...       foreignField: "faculty_id",
...       as: "courses_taught"
...     }
...   },
...   { $unwind: "$courses_taught" },
...
...   // Lookup enrollments for each course
...   {
...     $lookup: {
...       from: "enrollments",
...       localField: "courses_taught._id",
...       foreignField: "course_id",
...       as: "enrollments_info"
...     }
...   },
...   { $unwind: "$enrollments_info" },
...
...   // Lookup student details
...   {
...     $lookup: {
...       from: "students",
...       localField: "enrollments_info.student_id",
...       foreignField: "_id",
...       as: "student_info"
...     }
...   },
...   { $unwind: "$student_info" },
...
...   // Group by faculty and student to calculate average marks
...   {
...     $group: {
...       _id: { faculty: "$name", student: "$student_info.name" },
...       avg_marks: { $avg: "$enrollments_info.marks" }
...     }
...   },
...
...   // Group again by faculty to list all students
...   {
...     $group: {
...       _id: "$_id.faculty",
...       students: {
...         $push: {
```

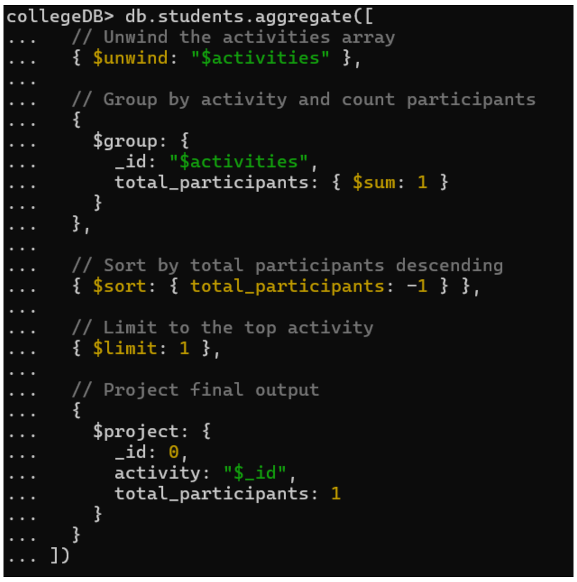


```
          name: "$_id.student",
          average_marks: "$avg_marks"
        }
      }
    }
  },

  // Project final output
  {
    $project: {
      _id: 0,
      faculty_name: "$_id",
      students: 1
    }
  }
])

students: [
  { name: 'Vincent Norris', average_marks: 86 },
  { name: 'David Taylor', average_marks: 65 }
],
faculty_name: 'Sandra Decker'
,

students: [
  { name: 'Kathryn Ferguson', average_marks: 84 },
  { name: 'Timothy Lee', average_marks: 85 },
  { name: 'Lydia Day', average_marks: 76 }
],
faculty_name: 'Robert Lara'
,

students: [ { name: 'Reginald Oliver', average_marks: 81 } ],
faculty_name: 'Ashley Parker'
,

students: [ { name: 'Carolyn Chandler', average_marks: 51 } ],
faculty_name: 'Charles Cunningham'
,

students: [
  { name: 'Rachael Harris', average_marks: 58 },
  { name: 'Ronald Trevino', average_marks: 97 }
],
faculty_name: 'Kevin Booth'
,

students: [
  { name: 'Lydia Day', average_marks: 53 },
  { name: 'Danielle Rich', average_marks: 68 },
  { name: 'Michelle Weber', average_marks: 71 }
],
faculty_name: 'Julia Cole'
,

students: [ { name: 'Ashley Myers', average_marks: 51 } ],
faculty_name: 'Stephen Galvan'
,

students: [ { name: 'Gabriela Le', average_marks: 55 } ],
faculty_name: 'Matthew Hanna'
,

students: [
  { name: 'Nicholas Turner', average_marks: 100 },
  { name: 'Juan Morris', average_marks: 80 }
```



```
  },
  {
    students: [ { name: 'Janet Jarvis', average_marks: 67 } ],
    faculty_name: 'Laura Flores'
  },
  {
    students: [ { name: 'Jeremy Carrillo', average_marks: 50 } ],
    faculty_name: 'Robin Johnson'
  },
  {
    students: [
      { name: 'Benjamin White', average_marks: 91 },
      { name: 'Thomas Jackson', average_marks: 82 },
      { name: 'Donna Spencer', average_marks: 81 }
    ],
    faculty_name: 'Jacqueline Miller'
  },
  {
    students: [ { name: 'Nicholas Turner', average_marks: 51 } ],
    faculty_name: 'Autumn White'
  },
  {
    students: [
      { name: 'Jason Brown', average_marks: 78 },
      { name: 'Alejandro Hart', average_marks: 65 },
      { name: 'Timothy Sparks', average_marks: 60 }
    ],
    faculty_name: 'James Martin'
  },
  {
    students: [
      { name: 'Marie Wilson', average_marks: 61 },
      { name: 'Travis Johnson', average_marks: 73 },
      { name: 'Christopher Benson', average_marks: 94 },
      { name: 'Thomas Jackson', average_marks: 85 }
    ],
    faculty_name: 'Kelly Huang'
  },
  {
    students: [
      { name: 'Barbara Jones', average_marks: 93 },
      { name: 'Anthony Zavala', average_marks: 90 }
    ],
    faculty_name: 'Alexis Stone'
  },
  {
    students: [
      { name: 'Lydia Day', average_marks: 92 },
      { name: 'Kyle Lee', average_marks: 97 }
    ],
    faculty_name: 'James Kirby'
  },
  {
    students: [
      { name: 'Timothy Sparks', average_marks: 80 },
      { name: 'Ryan Flores', average_marks: 84 }
    ],
    faculty_name: 'Cassandra Diaz'
  },
  {
    students: [
      { name: 'Timothy Sparks', average_marks: 95 },
      { name: 'Megan Taylor', average_marks: 74 }
    ],
    faculty_name: 'Robin West'
  }
]
Type "it" for more
```

Q14. Show the most popular activity type (e.g., Hackathon, Seminar, etc.) by number of student participants.



```
collegeDB> db.students.aggregate([
...   // Unwind the activities array
...   { $unwind: "$activities" },
...
...   // Group by activity and count participants
...   {
...     $group: {
...       _id: "$activities",
...       total_participants: { $sum: 1 }
...     }
...   },
...
...   // Sort by total participants descending
...   { $sort: { total_participants: -1 } },
...
...   // Limit to the top activity
...   { $limit: 1 },
...
...   // Project final output
...   {
...     $project: {
...       _id: 0,
...       activity: "$_id",
...       total_participants: 1
...     }
...   }
... ])
```