

# PROJECT

## 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".

**Solution:**

```
db.students.find( {  
    attendance: { $gt: 85 },  
    skills: { $all: ["MongoDB", "Python"] } },  
    {  
        _id: 0,  
        name: 1,  
        department: 1  
    });
```

**Output:**

```
mongo_project> db.students.find(  
...   // Name : sneha , registration no:1240258446  
... {  
...   attendance: { $gt: 85 },  
...   skills: { $all: ["MongoDB", "Python"] }  
... },  
... {  
...   _id: 0,  
...   name: 1,  
...   department: 1  
... }  
... );  
[ { name: 'Steven Wong' , department: 'Biotechnology' } ]
```

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

**Solution:**

```
db.faculty.aggregate(  
[ {  
    $project: {
```

```

        name: 1,
        totalCourses: { $size: "$courses" }
    } },
    {
        $match: { totalCourses: { $gt: 2 } }
    });

```

**Solution:**

```

[ { name: 'Steven Wong', department: 'Biotechnology' } ]
mongo_project> db.faculty.aggregate(
...   // Name : sneha , registration no:1240258446
...   [ {
...     $project: {
...       name: 1,
...       totalCourses: {
...         $size: "$courses" }
...     },
...     {
...       $match: {
...         totalCourses: { $gt: 2 } }
...     }
...   ]);
[
  { _id: 'F029', name: 'Charles Newton', totalCourses: 3 },
  { _id: 'F032', name: 'Julia Cole', totalCourses: 3 },
  { _id: 'F040', name: 'Darrell Velasquez', totalCourses: 3 },
  { _id: 'F048', name: 'Michael Poole', totalCourses: 3 },
  { _id: 'F051', name: 'John Duran', totalCourses: 3 },
  { _id: 'F061', name: 'Daniel Allen', totalCourses: 3 },
  { _id: 'F083', name: 'Matthew Hanna', totalCourses: 3 },
  { _id: 'F084', name: 'Michael Johnson', totalCourses: 3 },
  { _id: 'F100', name: 'Robert Lara', totalCourses: 3 }
]

```

## 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).

**Solution:**

```
db.enrollments.aggregate( [
```

```

{
  $lookup: {
    from: "students",
    localField: "student_id",
    foreignField: "_id",
    as: "studentInfo"
  }
},
{
  $lookup: {
    from: "courses",
    localField: "course_id",
    foreignField: "_id",
    as: "courseInfo"
  }
},
{
  $project: {
    _id: 0,
    studentName: { $arrayElemAt: ["$studentInfo.name", 0] },
    courseTitles: "$courseInfo.title"
  }
}
];
);

```

**Output:**

```
mongo_project> db.enrollments.aggregate( [  
...   // Name : sneha , registration no:1240258446  
... {  
...   $lookup: {  
...     from: "students",  
...     localField: "student_id",  
...     foreignField: "_id",  
...     as: "studentInfo"  
...   }  
... },  
... {  
...   $lookup: {  
...     from: "courses",  
...     localField: "course_id",  
...     foreignField: "_id",  
...     as: "courseInfo"  
...   }  
... },  
... {  
...   $project: {  
...     _id: 0,  
...     studentName: { $arrayElemAt: ["$studentInfo.name", 0] },  
...     courseTitles: "$courseInfo.title"  
...   }  
... }  
... ]  
... );  
[  
  {  
    studentName: 'Alexandra Bailey',  
    courseTitles: [ 'Reactive neutral adapter' ]  
  },  
  {  
    studentName: 'Megan Taylor',  
    courseTitles: [ 'Sharable bifurcated paradigm' ]  
  },  
  {  
    studentName: 'Alejandro Hart',  
    courseTitles: [ 'Focused user-facing paradigm' ]  
  },  
  {
```

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

**Solution:**

```
db.enrollments.aggregate( [
  {
    $group: {
      _id: "$course_id",
      totalStudents: { $sum: 1 },
      averageMarks: { $avg: "$marks" }
    }
  },
  {
    $lookup: {
      from: "courses",
      localField: "_id",
      foreignField: "_id",
      as: "courseInfo"
    }
  },
  {
    $project: {
      _id: 0,
      courseTitle: { $arrayElemAt: ["$courseInfo.title", 0] },
      totalStudents: 1,
      averageMarks: 1
    }
  }
]);

```

**Output:**

```
        averageMarks: 52,
        courseTitle: 'Automated global conglomeration'
    },
{
    totalStudents: 2,
    averageMarks: 77.5,
    courseTitle: 'Configurable scalable data-warehouse'
},
{
    totalStudents: 2,
    averageMarks: 75,
    courseTitle: 'Digitized even-keeled Internet solution'
},
{
    totalStudents: 1,
    averageMarks: 73,
    courseTitle: 'Innovative hybrid concept'
},
{
    totalStudents: 1,
    averageMarks: 81,
    courseTitle: 'User-centric upward-trending functionalities'
},
{
    totalStudents: 1,
    averageMarks: 86,
    courseTitle: 'Advanced analyzing budgetary management'
},
{
    totalStudents: 1,
    averageMarks: 92,
    courseTitle: 'Innovative mobile process improvement'
},
{
    totalStudents: 1,
    averageMarks: 74,
    courseTitle: 'Proactive optimizing initiative'
},
```

```

mongo_project> db.enrollments.aggregate( [
... // Name:sneha , registration no:1240258446
... {
... $group: { _id: "$course_id",
... totalStudents: { $sum: 1 },
... averageMarks: { $avg: "$marks" }
... },
... {
... $lookup: {
... from: "courses",
... localField: "_id",
... foreignField: "_id",
... as: "courseInfo"
... },
... {
... $project: {
... _id: 0,
... courseTitle: { $arrayElemAt: ["$courseInfo.title", 0] },
... totalStudents: 1,
... averageMarks: 1
... }
... ]
... );
[ {
... totalStudents: 1,
... averageMarks: 91,
... courseTitle: 'Decentralized multimedia Local Area Network'
},
{
... totalStudents: 2,
... averageMarks: 64.5,
... courseTitle: 'Balanced non-volatile parallelism'
},
{
... totalStudents: 1,
... averageMarks: 64,
... courseTitle: 'Fully-configurable empowering data-warehouse'
},
{
... totalStudents: 1,
...

```

### 3. Grouping, Sorting, and Limiting:

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

**Solution:**

```
db.enrollments.aggregate( [
```

```
{
```

```
$group: {
```

```
    _id: "$student_id",
```

```
averageMarks: { $avg: "$marks" }  
},  
{  
$sort: { averageMarks: -1 }  
,  
{  
$limit:3  
,  
{ $lookup: {  
from: "students",  
localField: "_id",  
foreignField: "_id",  
as: "studentInfo"  
}},  
{ $project: {  
_id: 0,  
studentName: { $arrayElemAt: ["$studentInfo.name", 0] },  
averageMarks: 1  
}  
}  
]  
);
```

**Output:**

```

mongo_project> db.enrollments.aggregate( [
...   // Name: sneha , registration no:1240258446
... {
...   $group: {
...     _id: "$student_id",
...     averageMarks: { $avg: "$marks" }
...   },
...   {
...     $sort: { averageMarks: -1 }
...   },
...   {
...     $limit:3
...   },
...   {
...     $lookup: {
...       from: "students",
...       localField: "_id",
...       foreignField: "_id",
...       as: "studentInfo"
...     },
...     { $project: {
...       _id: 0,
...       studentName: { $arrayElemAt: ["$studentInfo.name", 0] },
...       averageMarks: 1
...     } }
...   }
... ],
[ { averageMarks: 100, studentName: 'Diane Phillips' },
{ averageMarks: 98, studentName: 'Brandon Rios' },
{ averageMarks: 94, studentName: 'Larry Ramsey' }
]
)
mongo_project> |

```

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

**Solution:**

```

db.students.aggregate( [
{
  $group: {
    _id: "$department",
    totalStudents: { $sum: 1 }
  },
{
  $sort: { totalStudents: -1 }
},
{
  $limit: 1
}
]
)

```

```
    },
    {
      $project: {
        _id: 0,
        department: "$_id",
        totalStudents: 1
      }
    ]
  );
}
```

## Output

```
mongo_project> db.students.aggregate( [
...   // Name: sneha , registration no:1240258446
...   {
...     $group: {
...       _id: "$department",
...       totalStudents: { $sum: 1
...     }
...   },
...   {
...     $sort: { totalStudents: -1 }
...   },
...   {
...     $limit: 1
...   },
...   {
...     $project: {
...       _id: 0,
...       department: "$_id",
...       totalStudents: 1
...     }
...   }
... ]
... );
[ { totalStudents: 23, department: 'Electrical' } ]
```

## **4. Update, Insert, and Delete:**

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

### Solution:

```
db.students.updateMany(  
  {  
    activities: "Hackathon"  
  },  
  {  
    $set: { attendance: 100 }  
  }  
);
```

**Output:**

```
mongo_project> db.students.updateMany(  
... // Name: sneha , registration no:1240258446  
... {  
...   activities: "Hackathon"  
... },  
... {  
...   $set: { attendance: 100 }  
... }  
... );  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 0,  
  modifiedCount: 0,  
  upsertedCount: 0  
}
```

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

**Solution:**

```
db.activities.deleteMany(  
  {  
    year: { $lt: 2022 }  
  }  
);
```

**Output:**

```
mongo_project> db.activities.deleteMany(  
... // Name: sneha , registration no:1240258446  
... {  
... year: { $lt: 2022 }  
... }  
... );  
{ acknowledged: true, deletedCount: 0 }
```

**Q9.** Insert 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".

**Solution:**

```
db.courses.updateOne(  
{ _id: "C150" },  
{  
$set: { title: "Advanced Data Structures", credits: 4 }  
},  
{ upsert: true }  
);
```

**Output:**

```
mongo_project> db.courses.updateOne(  
... // Name: sneha , registration no:1240258446  
... { _id: "C150" },  
... {  
$set: { title: "Advanced Data Structures", credits: 4 }  
},  
{ upsert: true }  
);  
{  
acknowledged: true,  
insertedId: 'C150',  
matchedCount: 0,  
modifiedCount: 0,  
upsertedCount: 1  
}
```

## 5. Array & Operator Usage:

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

**Solution:**

```
db.students.find(  
{  
  $and: [  
    { skills: "Python" },  
    { skills: { $ne: "C++" } }  
  ],  
  {  
    _id: 0,  
    name: 1,  
    skills: 1  
  }  
);
```

## Output:

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

**Solution:**

```
db.activities.aggregate([
  {
    $group: {
      _id: "$student_id",
      activities: { $addToSet: "$type" }
    },
    {
      $match: {
        activities: { $all: ["Seminar", "Hackathon"] }
      },
      {
        $lookup: {
          from: "students",
          localField: "_id",
          foreignField: "_id",
          as: "student_details"
        }
      },
      {
        $project: {
          _id: 0,
          name: { $arrayElemAt: ["$student_details.name", 0] }
        } });
  }
]);
```

**Output:**

```

mongo_project> db.activities.aggregate([
... // Name:sneha , registration no:1240258446
... {
... $group: {
... _id: "$student_id",
... activities: { $addToSet: "$type" }
... },
... {
... $match: {
... activities: { $all: ["Seminar", "Hackathon"] }
... },
... {
... $lookup: {
... from: "students",
... localField: "_id",
... foreignField: "_id",
... as: "student_details"
... },
... {
... $project: {
... _id: 0,
... name: { $arrayElemAt: ["$student_details.name", 0] }
... }
... ]);
[{"name": "Lydia Day"}, {"name": "Taylor Webb"}, {"name": "Patricia Scott"}, {"name": "Carlos Bryant"}, {"name": "Adam Solomon"}]
]

```

## 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.

**Solution:**

```

db.enrollments.aggregate([
//Name- Sneha, University Roll No- 1240258446
{
  $lookup: {
    from: "students",
    localField: "student_id",
    foreignField: "_id",
    as: "student_info"
  }
  $match: {
    student_info: {
      $elemMatch: {
        department: "Computer Science"
      }
    }
  }
  $project: {
    _id: 0,
    name: 1,
    score: 1
  }
  $sort: {
    score: -1
  }
}
])

```

```

        }
    },
    { $unwind: "$student_info" },
    {
        $lookup: {
            from: "courses",
            localField: "course_id",
            foreignField: "_id",
            as: "course_info"
        }
    },
    { $unwind: "$course_info" },
    {
        $match: {
            "course_info.title": "Web development",
            "student_info.department": "Computer Science",
            marks: { $gt: 80 }
        }
    },
    {
        $project: {
            _id: 0,
            student_name: "$student_info.name",
            department: "$student_info.department",
            course: "$course_info.title",
            marks: 1
        }
    }
]);

```

**Output:**

```

MongoProject> db.enrollments.aggregate([
... //Name- Sneha, University Roll No- 1240258446
... {
...   $lookup: {
...     from: "students",
...     localField: "student_id",
...     foreignField: "_id",
...     as: "student_info"
...   }
... },
... {$unwind: "$student_info" },
... {
...   $lookup: {
...     from: "courses",
...     localField: "course_id",
...     foreignField: "_id",
...     as: "course_info"
...   }
... },
... {$unwind: "$course_info" },
... {
...   $match: {
...     "course_info.title": "Web development",
...     "student_info.department": "Computer Science",
...     marks: { $gt: 80 }
...   }
... },
... {
...   $project: {
...     _id: 0,
...     student_name: "$student_info.name",
...     department: "$student_info.department",
...     course: "$course_info.title",
...     marks: 1
...   }
... }
... ]);
...
MongoProject>

```

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

Solution:

```

db.faculty.aggregate([
{
  $lookup: {
    from: "courses",
    localField: "_id",
    foreignField: "faculty_id",
    as: "course_info"
  }
},

```

```
{ $unwind: "$course_info" },
{
$lookup: {
from: "enrollments",
localField: "course_info._id",
foreignField: "course_id",
as: "enroll_info"
} },
{ $unwind: "$enroll_info" },
{
$lookup: {
from: "students",
localField: "enroll_info.student_id",
foreignField: "_id",
as: "student_info"
} },
{ $unwind: "$student_info" },
{
$group: {
_id: {
faculty_id: "$_id",
faculty_name: "$name",
student_name: "$student_info.name"
},
avg_marks: { $avg: "$enroll_info.marks" }
} },
{
$group: {
_id: {
faculty_id: "$_id.faculty_id",
faculty_name: "$_id.faculty_name" }
},
```

```
students: {  
    $push: {  
        student_name: "$_id.student_name",  
        average_marks: { $round: ["$avg_marks", 2] }  
    } } },  
{ $project: {  
    _id: 0,  
    faculty_name: "$_id.faculty_name",  
    students: 1  
} ] );
```

Output:

```
mongoProject> db.faculty.aggregate([
... //Name- Sneha, University Roll No- 1240258446
... {
...   $lookup: {
...     from: "courses",
...     localField: "_id",
...     foreignField: "faculty_id",
...     as: "course_info"
...   },
...   {$unwind: "$course_info" },
...   {
...     $lookup: {
...       from: "enrollments",
...       localField: "course_info._id",
...       foreignField: "course_id",
...       as: "enroll_info"
...     },
...     {$unwind: "$enroll_info" },
...     {
...       $lookup: {
...         from: "students",
...         localField: "enroll_info.student_id",
...         foreignField: "_id",
...         as: "student_info"
...       },
...       {$unwind: "$student_info" },
...       {
...         $group: {
...           _id: {
...             faculty_id: "$_id",
...             faculty_name: "$name",
...             student_name: "$student_info.name"
...           },
...           avg_marks: { $avg: "$enroll_info.marks" }
...         },
...         $group: {
...           _id: {
...             faculty_id: "$_id.faculty_id",
...             faculty_name: "$_id.faculty_name"
...           },
...           students: {
...             $push: {
...               student_name: "$_id.student_name",
...             }
...           }
...         }
...       }
...     }
...   }
... ])
```

```

},
  students: {
    $push: {
      student_name: "$_id.student_name",
      average_marks: { $round: ["$avg_marks", 2] }
    }
  },
  { $project: {
    _id: 0,
    faculty_name: "$_id.faculty_name",
    students: 1
  } ] );
...
[
  {
    students: [ { student_name: 'Brian Russell', average_marks: 82 } ],
    faculty_name: 'Darrell Velasquez'
  },
  {
    students: [
      { student_name: 'Aaron Marshall', average_marks: 58 },
      { student_name: 'Danielle Rich', average_marks: 60 },
      { student_name: 'Larry Ramsey', average_marks: 94 }
    ],
    faculty_name: 'Daniel Allen'
  },
  {
    students: [ { student_name: 'Carolyn Chandler', average_marks: 51 } ],
    faculty_name: 'Charles Cunningham'
  },
  {
    students: [ { student_name: 'Bruce Blair', average_marks: 78 } ],
    faculty_name: 'Jessica Black'
  },
  {
    students: [
      { student_name: 'Benjamin White', average_marks: 91 },
      { student_name: 'Donna Spencer', average_marks: 81 },
      { student_name: 'Thomas Jackson', average_marks: 82 }
    ],
    faculty_name: 'Jacqueline Miller'
  },
  {
    students: [ { student_name: 'Michelle Walters', average_marks: 91 } ],
    faculty_name: 'Julie Elliott'
  }
]

```

sQ14. Show the most popular activity type (e.g., Hackathon, Seminar, etc.) by number of student participants.

Solution:

```

db.activities.aggregate([
  {
    $group: {
      _id: "$type",
      totalParticipants: { $addToSet: "$student_id" }
    }
  }
])

```

```
{
  $project: {
    _id: 1,
    participantCount: { $size: "$totalParticipants" }
  },
  {
    $sort: { participantCount: -1 }
  },
  {
    $limit: 1
  } ] );
}
```

**Outpr:**

```
MongoProject> db.activities.aggregate([
... //Name- Sneha, University Roll No- 1240258446
...   {
...     $group: {
...       _id: "$type",
...       totalParticipants: { $addToSet: "$student_id" }
...     },
...     {
...       $project: {
...         _id: 1,
...         participantCount: { $size: "$totalParticipants" }
...       },
...       {
...         $sort: { participantCount: -1 }
...       },
...       {
...         $limit: 1
...       } ] );
...
[ { _id: 'Hackathon', participantCount: 29 } ]
```