

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(  
  //Name- Sneha Singh, University Roll No- 1240258448  
  {  
    attendance: { $gt: 85 },  
    skills: { $all: ["MongoDB", "Python"] }  
  },  
  {  
    _id: 0,  
    name: 1,  
    department: 1  
  }  
);
```

Output:-

```
MongoProject> db.students.find(  
... //Name- Sneha Singh, University Roll No- 1240258448  
... {  
...   attendance: { $gt: 85 },  
...   skills: { $all: ["MongoDB", "Python"] }  
... },  
... {  
...   _id: 0,  
...   name: 1,  
...   department: 1  
... }  
... );  
...  
[ { name: 'Steven Wong', department: 'Biotechnology' } ]  
MongoProject> |
```

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([
```

```
//Name- Sneha Singh, University Roll No- 1240258448
```

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

Output:-

```
MongoProject> db.faculty.aggregate([
... //Name- Sneha Singh, University Roll No- 1240258448
... {
...   $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([
//Name- Sneha Singh, University Roll No- 1240258448
{
  $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" },
    {
      $project: {
        _id: 0,
        student_name: "$student_info.name",
        course_title: "$course_info.title"
      }
    }
  }
] );
```

```

MongoProject> db.enrollments.aggregate([
...   //Name- Sneha Singh, University Roll No- 1240258448
...   {
...     $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" },
...     { $project: {
...       _id: 0,
...       student_name: "$student_info.name",
...       course_title: "$course_info.title"
...     } } ] );
...
[
  {
    student_name: 'Alexandra Bailey',
    course_title: 'Reactive neutral adapter'
  },
  {
    student_name: 'Megan Taylor',
    course_title: 'Sharable bifurcated paradigm'
  },
  {
    student_name: 'Alejandro Hart',
    course_title: 'Focused user-facing paradigm'
  },
  {
    student_name: 'Timothy Sparks',
    course_title: 'Focused user-facing paradigm'
  },
  {
    student_name: 'Juan Morris',
    course_title: 'Balanced asynchronous framework'
  },
  {
    student_name: 'Donna Morgan',
    course_title: 'Organic optimal product'
  },
  {
    student_name: 'Patricia Scott',
    course_title: 'Fully-configurable responsive solution'
  }
]

```

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

Solution:

```
db.enrollments.aggregate( [  
  //Name- Sneha Singh, University Roll No- 1240258448  
  {  
    $group: {  
      _id: "$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  
    }  
  }  
]);
```

```

MongoProject> db.enrollments.aggregate( [
...   //Name- Sneha Singh, University Roll No- 1240258448
...   {
...     $group: {
...       _id: "$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: 2,
    averageMarks: 94.5,
    courseTitle: 'Total tangible moderator'
  },
  {
    totalStudents: 1,
    averageMarks: 67,
    courseTitle: 'Focused multi-state encoding'
  },
  {
    totalStudents: 3,
    averageMarks: 92.33333333333333,
    courseTitle: 'Persevering asynchronous hub'
  },
  {
    totalStudents: 1,
    averageMarks: 67,
    courseTitle: 'Configurable global framework'
  },
  {
    totalStudents: 1,
    averageMarks: 92,
    courseTitle: 'Innovative mobile process improvement'
  },
]

```

3. Grouping, Sorting, and Limiting

Q.5 Find the top 3 students with the highest average marks across all enrolled courses

Solution:

```
db.enrollments.aggregate([
    //Name- Sneha Singh, University Roll No- 1240258448
    {
        $group: {
            _id: "$student_id",
            averageMarks: { $avg: "$marks" }
        } },
    {
        $lookup: {
            from: "students",
            localField: "_id",
            foreignField: "_id",
            as: "student_info"
        } },
    { $unwind: "$student_info" },
    {
        $project: {
            _id: 0,
            student_id: "$_id",
            name: "$student_info.name",
            department: "$student_info.department",
            averageMarks: { $round: ["$averageMarks", 2] }
        } },
    { $sort: { averageMarks: -1 } },
    { $limit: 3 }
])
```

```

    { $sort: { averageMarks: -1 } },
    { $limit: 3 }
  ] );

```

Output:-

```

MongoProject> db.enrollments.aggregate([
...   //Name- Sneha Singh, University Roll No- 1240258448
...   {
...     $group: {
...       _id: "$student_id",
...       averageMarks: { $avg: "$marks" }
...     },
...     {
...       $lookup: {
...         from: "students",
...         localField: "_id",
...         foreignField: "_id",
...         as: "student_info"
...       },
...       { $unwind: "$student_info" },
...       $project: {
...         _id: 0,
...         student_id: "$_id",
...         name: "$student_info.name",
...         department: "$student_info.department",
...         averageMarks: { $round: ["$averageMarks", 2] }
...       },
...       { $sort: { averageMarks: -1 } },
...       { $limit: 3 }
...     ] );
...
... \[
... {
...   student_id: 'S080',
...   name: 'Diane Phillips',
...   department: 'Civil',
...   averageMarks: 100
... },
... {
...   student_id: 'S046',
...   name: 'Brandon Rios',
...   department: 'Biotechnology',
...   averageMarks: 98
... },
... {
...   student_id: 'S041',
...   name: 'Christopher Benson',
...   department: 'Electrical',
...   averageMarks: 94
... }
... ]

```


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

Solution:

```
db.students.aggregate([
    //Name- Sneha Singh, University Roll No- 1240258448
    {
        $group: {
            _id: "$department",
            totalStudents: { $sum: 1 }
        }
    },
    {
        $sort: { totalStudents: -1 }
    },
    { $limit: 1 }
])
```

Output:-

```
MongoProject> db.students.aggregate([
... //Name- Sneha Singh, University Roll No- 1240258448
... {
...   $group: {
...     _id: "$department",
...     totalStudents: { $sum: 1 }
...   }
... },
... {
...   $sort: { totalStudents: -1 }
... },
... { $limit: 1 }
... ] );
...
[ { _id: 'Electrical', totalStudents: 23 } ]
```

4. Update, Upsert, and Delete

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

Solution:

```
db.students.updateMany(
    //Name- Sneha Singh, University Roll No- 1240258448
    {
        _id: {
            $in: db.activities.distinct("student_id", { type: "Hackathon",
            position: "Winner" })
        }
    },
    { $set: { attendance: 100 } }
);
```

Output:-

```
MongoProject> db.students.updateMany(
...     //Name- Sneha Singh, University Roll No- 1240258448
...     {
...         _id: {
...             $in: db.activities.distinct("student_id", { type: "Hackathon", position: "Winner" })
...         }
...     },
...     { $set: { attendance: 100 } }
... );
...
... {
...     acknowledged: true,
...     insertedId: null,
...     matchedCount: 9,
...     modifiedCount: 9,
...     upsertedCount: 0
... }
```

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

Solution:

```
db.activities.deleteMany(  
  //Name- Sneha Singh, University Roll No- 1240258448  
  { year: { $lt: 2022 } }  
);
```

Output:-

```
MongoProject> db.activities.deleteMany(  
...   //Name- Sneha Singh, University Roll No- 1240258448  
...   { year: { $lt: 2022 } }  
... );  
...  
{ acknowledged: true, deletedCount: 0 }
```

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".

Solution:

```
db.courses.updateOne(  
  //Name- Sneha Singh, University Roll No- 1240258448  
  
  { _id: "C150" },  
  { $set: { title: "Advanced Data Structures", credits: 4 } },  
  { upsert: true }  
);
```

Output:-

```
MongoProject> db.courses.updateOne(  
...   //Name- Sneha Singh, University Roll No- 1240258448  
...  
...   { _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(
//Name- Sneha Singh, University Roll No- 1240258448
{
  skills: { $all: ["Python"], $ne: "C++" }
}
);
```

Output:-

```
MongoProject> db.students.find(
... //Name- Sneha Singh, University Roll No- 1240258448
... {
...   skills: { $all: ["Python"], $ne: "C++" }
... }
... );
...
...
[
  {
    _id: 'S004',
    name: 'Kyle Hale',
    dob: '2000-10-20',
    department: 'Electrical',
    skills: [ 'Python', 'Java' ],
    attendance: 79.78
  },
  {
    _id: 'S008',
    name: 'Cody Whitehead',
    dob: '2003-11-25',
    department: 'Biotechnology',
    skills: [ 'JavaScript', 'Python' ],
    attendance: 100
  },
  {
    _id: 'S009',
    name: 'Thomas Jackson',
    dob: '2002-10-25',
    department: 'Electrical',
    skills: [ 'Python', 'AutoCAD' ],
    attendance: 96.64
  },
  {
    _id: 'S012',
    name: 'Steven Wong',
    dob: '2003-09-06',
    department: 'Biotechnology',
    skills: [ 'MongoDB', 'Python' ],
    attendance: 100
  },
  {
    _id: 'S015',
```

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

Solution:

```
db.activities.aggregate([
  //Name- Sneha Singh, University Roll No- 1240258448
  {
    $group: {
      _id: "$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:-

```
MongoProject> db.activities.aggregate([
...   //Name- Sneha Singh, University Roll No- 1240258448
...   {
...     $group: {
...       _id: "$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: 'Adam Solomon' },
  { name: 'Lydia Day' },
  { name: 'Carlos Bryant' },
  { name: 'Taylor Webb' },
  { name: 'Patricia Scott' }
]
```

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 Singh, University Roll No- 1240258448
```

```
  
  {  
    $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
    }
  }
]);

```

Output:-

```

MongoProject> db.enrollments.aggregate([
... //Name- Sneha Singh, University Roll No- 1240258448
...
... {
...   $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
...   }
... }
... ]);
...

```


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.

Solution:

```
db.faculty.aggregate([
  //Name- Sneha Singh, University Roll No- 1240258448
  {
    $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 Singh, University Roll No- 1240258448
... {
...   $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: {
```

```
... { $project: {
...   _id: 0,
...   faculty_name: "$_id.faculty_name",
...   students: 1
... } } ];
...
... {
...   students: [
...     { student_name: 'Ronald Trevino', average_marks: 93 },
...     { student_name: 'Tracey Young', average_marks: 67 }
...   ],
...   faculty_name: 'Maxwell Harrison'
... },
... {
...   students: [
...     { student_name: 'Ariana Ward', average_marks: 91 },
...     { student_name: 'Joseph Brown', average_marks: 52 }
...   ],
...   faculty_name: 'Michele Hines'
... },
... {
...   students: [
...     { student_name: 'Vincent Norris', average_marks: 86 },
...     { student_name: 'David Taylor', average_marks: 65 }
...   ],
...   faculty_name: 'Sandra Decker'
... },
... {
...   students: [
...     { student_name: 'Thomas Jackson', average_marks: 82 },
...     { student_name: 'Benjamin White', average_marks: 91 },
...     { student_name: 'Donna Spencer', average_marks: 81 }
...   ],
...   faculty_name: 'Jacqueline Miller'
... },
... {
...   students: [
...     { student_name: 'Christopher Benson', average_marks: 94 },
...     { student_name: 'Marie Wilson', average_marks: 61 },
...     { student_name: 'Travis Johnson', average_marks: 73 },
...     { student_name: 'Thomas Jackson', average_marks: 85 }
...   ],
...   faculty_name: 'Kelly Huang'
... },
... {
...   students: [ { student_name: 'Jessica Galvan', average_marks: 64 } ],
...   faculty_name: 'Kathryn Young'
... },
... ]
```

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

Solution:

```
db.activities.aggregate([
    //Name- Sneha Singh, University Roll No- 1240258448
    {
        $group: {
            _id: "$type",
            totalParticipants: { $addToSet: "$student_id" }
        },
        {
            $project: {
                _id: 1,
                participantCount: { $size: "$totalParticipants" }
            },
            {
                $sort: { participantCount: -1 }
            },
            {
                $limit: 1
            }
        ]
    });
```

Output:-

```
MongoProject> db.activities.aggregate([
... //Name- Sneha Singh, University Roll No- 1240258448
... {
...   $group: {
...     _id: "$type",
...     totalParticipants: { $addToSet: "$student_id" }
...   },
...   {
...     $project: {
...       _id: 1,
...       participantCount: { $size: "$totalParticipants" }
...     },
...     {
...       $sort: { participantCount: -1 }
...     },
...     {
...       $limit: 1
...     }
...   }
... ] );
...
[ { _id: 'Hackathon', participantCount: 29 } ]
MongoProject>
```