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:

```
university> db.students.find(
...   // Name: Himanshu Gautam Registration No: 1240258198
...   {
...     attendance: { $gt: 85 },
...     skills: { $all: ["MongoDB", "Python"] }
...   },
...   {
...     _id: 0,
...     name: 1,
...     department: 1
...   }
... );
...

university>
```

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
} }
  }]);
```

Output:

```
university> db.faculty.aggregate(
...   // Name: Himanshu Gautam Registration No: 1240258198
...   [
...     {
...       $project: {
...         name: 1,
...         totalCourses: { $size: "$courses" }
...       }
...     },
...     {
...       $match: { totalCourses: { $gt: 2 } }
...     }
...   ]
... );
...

university>
```

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:

```
university> db.enrollments.aggregate(
...   // Name: Himanshu Gautam Registration No: 1240258198
...   [
...     {
...       $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:

```
university> db.enrollments.aggregate(
...   // Name: Himanshu Gautam Registration No: 1240258198
...   [
...     {
...       $group: {
...         _id: "$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: 86.5,
    courseTitle: 'Fully-configurable responsive solution'
  },
  {
    totalStudents: 4,
    averageMarks: 82.5,
    courseTitle: 'Customizable client-driven secured line'
  }
```

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:

```
university> db.enrollments.aggregate(
...   // Name: Himanshu Gautam Registration No: 1240258198
...   [
...     {
...       $group: {
...         _id: "$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' }
]
university>
fwd-i-search: _
```

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

Solution: **Output:**

```
db.students.aggregate( [
   {
     $group: {
       _id: "$department",
       totalStudents: {
$sum: 1 }
     }
   },
   {
     $sort: { totalStudents:
-1 }
   },
   {
     $limit: 1
   },
   {
     $project: {
       _id: 0,
       department: "$_id",
       totalStudents: 1
     }
   }
 ]
);
```

```
university> db.students.aggregate(
...   // Name: Himanshu Gautam Registration No: 1240258198
...   [
...     {
...       $group: {
...         _id: "$department",
...         totalStudents: { $sum: 1 }
...       }
...     },
...     { $sort: { totalStudents: -1 } },
...     { $limit: 1 },
...     {
...       $project: {
...         _id: 0,
...         department: "$_id",
...         totalStudents: 1
...       }
...     }
...   ]
... );
...
[ { totalStudents: 23, department: 'Electrical' } ]
university>
```

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:

```
university> db.students.updateMany(
...   // Name: Himanshu Gautam Registration No: 1240258198
...   {
...     activities: "Hackathon"
...   },
...   {
...     $set: { attendance: 100 }
...   }
... );
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
university>
```

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

Solution:

```
db.activities.deleteMany(
  {

    year: { $lt: 2022 }
  }
);
```

Output:

```
university> db.activities.deleteMany(
...   // Name: Himanshu Gautam Registration No: 1240258198
...   {
...     year: { $lt: 2022 }
...   }
... );
...
{ acknowledged: true, deletedCount: 0 }
university>
```

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:

```
university> db.courses.updateOne(
...   // Name: Himanshu Gautam Registration No: 1240258198
...   { _id: "C150" },
...   {
...     $set: { title: "Advanced Data Structures", credits: 4 }
...   },
...   { upsert: true }
... );
...
{
  acknowledged: true,
  insertedId: 'C150',
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
university>
```

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:

```
university> db.students.find(
...   // Name: Himanshu Gautam Registration No: 1240258198
...   {
...     skills: "Python",
...     skills: { $nin: ["C++"] }
...   },
...   {
...     _id: 0,
...     name: 1,
...     skills: 1
...   }
... );
...
[
  { name: 'Bruce Blair', skills: [ 'MongoDB', 'Linux' ] },
  { name: 'Alexandra Bailey', skills: [ 'Research', 'AutoCAD' ] },
  { name: 'Kyle Hale', skills: [ 'Python', 'Java' ] },
  { name: 'Daniel Robinson', skills: [ 'JavaScript', 'Java' ] },
  { name: 'Tina Hodge', skills: [ 'SQL', 'Research' ] },
  { name: 'Anthony Zavala', skills: [ 'Java', 'Git' ] },
  { name: 'Cody Whitehead', skills: [ 'JavaScript', 'Python' ] },
  { name: 'Thomas Jackson', skills: [ 'Python', 'AutoCAD' ] },
  { name: 'Monica Martin', skills: [ 'Research', 'JavaScript' ] },
  { name: 'Kathryn Ferguson', skills: [ 'Java', 'Linux' ] },
  { name: 'Steven Wong', skills: [ 'MongoDB', 'Python' ] },
```

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

Solution:

```
db.students.find(
  {

    activities: { $all:
["Seminar", "Hackathon"]
}
  },
  {
    _id: 0,
    name: 1,
```

Output:

```
university> db.students.find(
...   // Name: Himanshu Gautam Registration No: 1240258198
...   {
...     activities: { $all: ["Seminar", "Hackathon"] }
...   },
...   {
...     _id: 0,
...     name: 1,
...     activities: 1
...   }
... );
...

university>
```

```
    activities: 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.

Solution:

```
db.enrollments.find(
{
  course_title: "Web Development",
  marks: { $gt: 80 },
  department: "Computer Science"
 },
 {
  _id: 0,
  student_id: 1,
  marks: 1,
  course_title: 1,
  department: 1
 }
);
```

Output:

```
university> db.enrollments.find(
...   // Name: Himanshu Gautam Registration No: 1240258198
...   {
...     course_title: "Web Development",
...     marks: { $gt: 80 },
...     department: "Computer Science"
...   },
...   {
...     _id: 0,
...     student_id: 1,
...     marks: 1,
...     course_title: 1,
...     department: 1
...   }
... );
...

university>
```

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( [
  {
   $lookup: {
    from: "courses",
    localField: "courses",
    foreignField: "_id",
    as: "courseInfo"
   }
  },
  {
   $group: {
    _id: {
facultyName: "$facultyName",
studentName: "$studentName"
},
```

```
},
{ $unwind: "$courseInfo" },
{
  $lookup: {
    from: "enrollments",
    localField: "courseInfo._id",
    foreignField: "course_id",
    as: "enrolledStudents"

  }
},
{ $unwind: "$enrolledStudents" },
{
  $lookup: {
    from: "students",
    localField: "enrolledStudents.student_id",
    foreignField: "_id",
    as: "studentInfo"
  }
},
{
  $project: {
    _id: 0,
    facultyName: "$name",
    studentName: { $arrayElemAt: ["$studentInfo.name",0] },
    marks: "$enrolledStudents.marks"
  }
averageMarks: {
$avg: "$marks" }
      }
    },
    {
      $project: {
        _id: 0,


facultyName:
"$_id.facultyNam
e",

studentName:
"$_id.studentNa
me",

averageMarks: 1
      }
    },
    {
      $sort: {
facultyName: 1,
studentName: 1
}
    }
  ]
);
```

Output:

```
university> db.faculty.aggregate(
...   // Name: Himanshu Gautam Registration No: 1240258198
...   [
...     {
...       $lookup: {
...         from: "courses",
...         localField: "courses",
...         foreignField: "_id",
...         as: "courseInfo"
...       }
...     },
...     { $unwind: "$courseInfo" },
...     {
...       $lookup: {
...         from: "enrollments",
...         localField: "courseInfo._id",
...         foreignField: "course_id",
...         as: "enrolledStudents"
...       }
...     },
...     { $unwind: "$enrolledStudents" },
...     {
...       $lookup: {
...         from: "students",
...         localField: "enrolledStudents.student_id",
...         foreignField: "_id",
...         as: "studentInfo"
...       }
...     },
...     {
...       $project: {
...         _id: 0,
...         facultyName: "$name",
...         studentName: { $arrayElemAt: ["$studentInfo.name", 0] },
...         marks: "$enrolledStudents.marks"
...       }
...     },
...     {
...       $group: {
...         _id: { facultyName: "$facultyName", studentName: "$studentName
...         averageMarks: { $avg: "$marks" }
```

```
...       $group: {
...         _id: { facultyName: "$facultyName", studentName: "$studentName" },
...         averageMarks: { $avg: "$marks" }
...       }
...     },
...     {
...       $project: {
...         _id: 0,
...         facultyName: "$_id.facultyName",
...         studentName: "$_id.studentName",
...         averageMarks: 1
...       }
...     },
...     { $sort: { facultyName: 1, studentName: 1 } }
...   ]
... );
...

university>
```

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

Solution:

```
db.students.aggregate( [
   { $unwind: "$activities" },
   {

    $group: {
      _id: "$activities",
      totalParticipants: { $sum: 1 }
    }
   },
   { $sort: { totalParticipants: -1 }
},
   { $limit: 1 },
   {
    $project: {
      _id: 0,
      activity: "$_id",
      totalParticipants: 1
    }
   }
 ]
);
```

Output:

```
university> db.students.aggregate(
...   // Name: Himanshu Gautam Registration No: 1240258198
...   [
...     { $unwind: "$activities" },
...     {
...       $group: {
...         _id: "$activities",
...         totalParticipants: { $sum: 1 }
...       }
...     },
...     { $sort: { totalParticipants: -1 } },
...     { $limit: 1 },
...     {
...       $project: {
...         _id: 0,
...         activity: "$_id",
...         totalParticipants: 1
...       }
...     }
...   ]
... );
...

university>
```