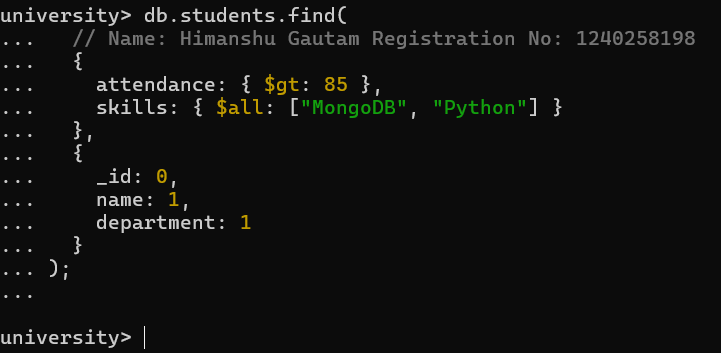
PROJECT

# 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"] }

},

{

## Output:

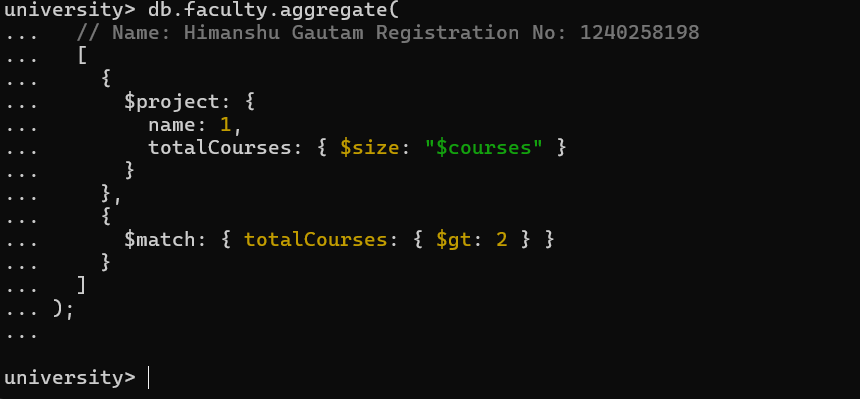
\_id: 0,

name: 1,

department: 1

}

);

**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" }

} },

## Output:

{

$match: {

totalCourses: { $gt: 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:

****

**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:**



# 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

}

}

]

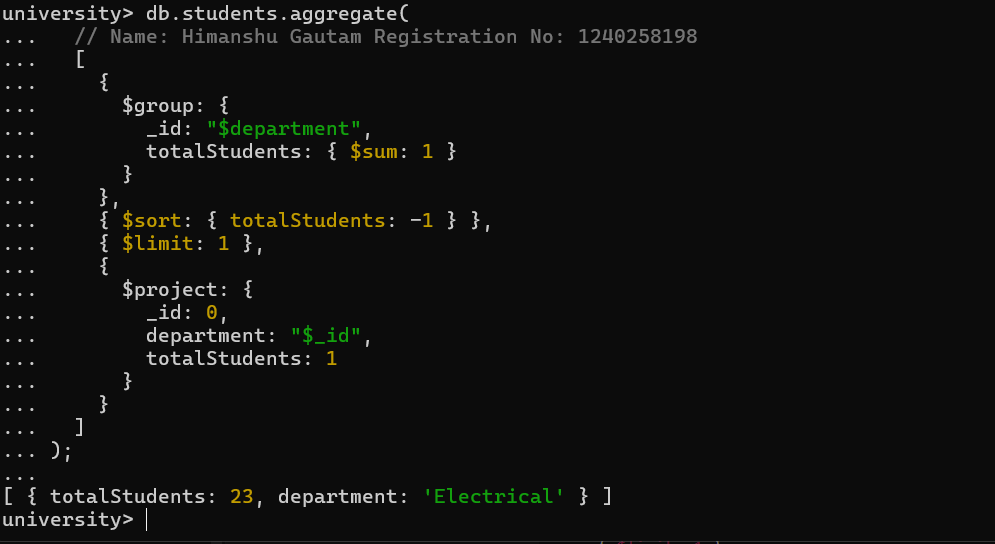
);

## Output:



**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

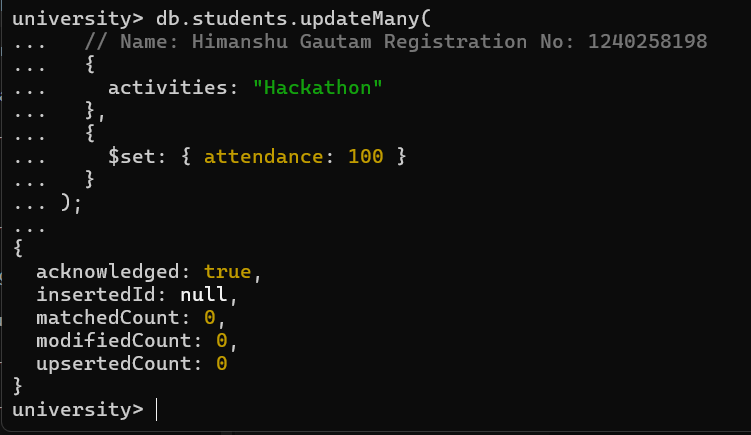
}

}

]

);

# Update, Insert, and Delete:

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

## Solution:

db.students.updateMany(

{

activities: "Hackathon"

},

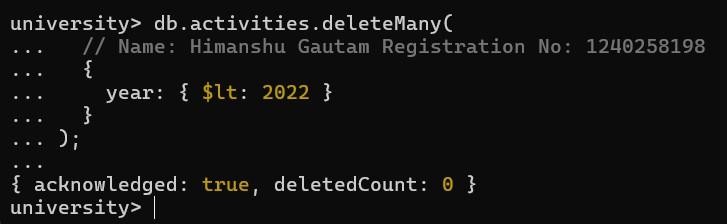
{

## Output:

$set: { attendance: 100 }

}

);

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

## Solution:

db.activities.deleteMany(

{

## Output:

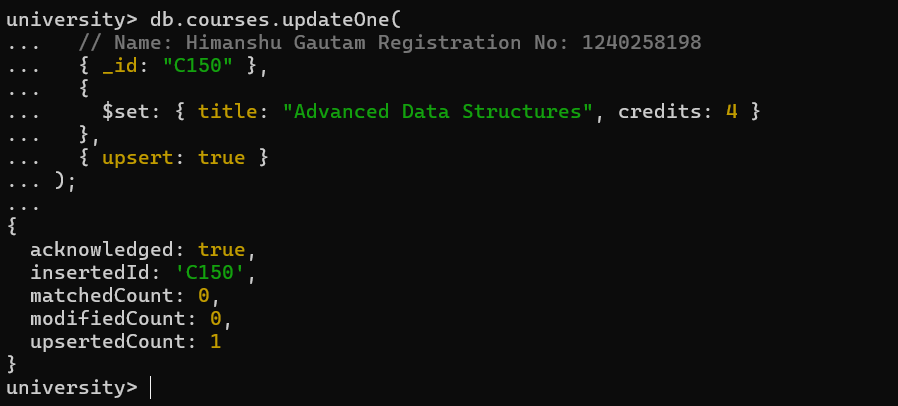
year: { $lt: 2022 }

}

);

**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:**

# 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++" } }

## Output:

]

},

{

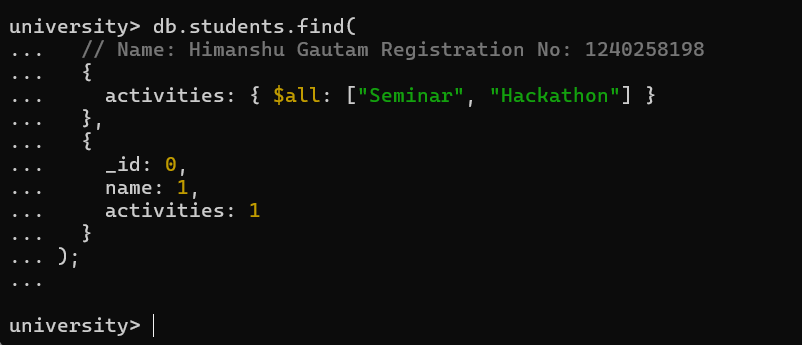
\_id: 0,

name: 1,

skills: 1

}

);

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

## Solution:

db.students.find(

{

activities: { $all: ["Seminar", "Hackathon"]

}

## Output:

},

{

\_id: 0,

name: 1,

activities: 1

}

);

# 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",

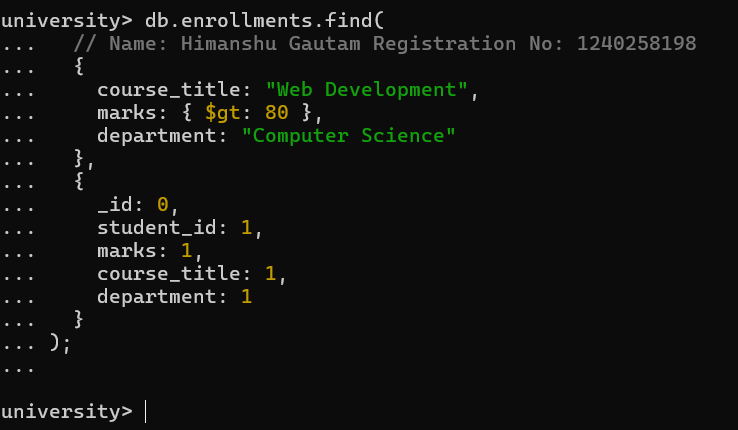
marks: { $gt: 80 }, department:

"Computer Science"

},

{

## Output:

\_id: 0,

student\_id: 1,

marks: 1,

course\_title: 1,

department: 1

}

);

# 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"

averageMarks: {

$avg: "$marks" }

}

},

{

$project: {

\_id: 0,

}

}, facultyName:

{ $unwind: "$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"

}

"$\_id.facultyNam e",

studentName: "$\_id.studentNa me",

averageMarks: 1

}

},

{

$sort: { facultyName: 1,

studentName: 1

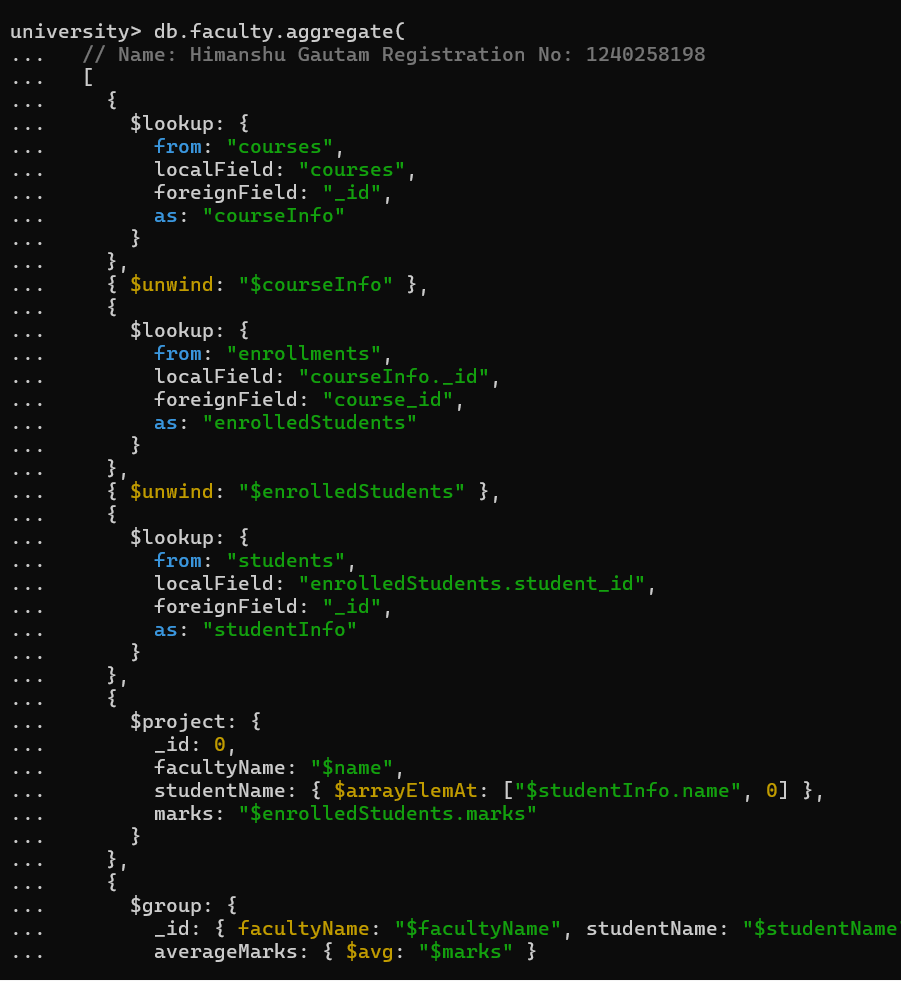
}

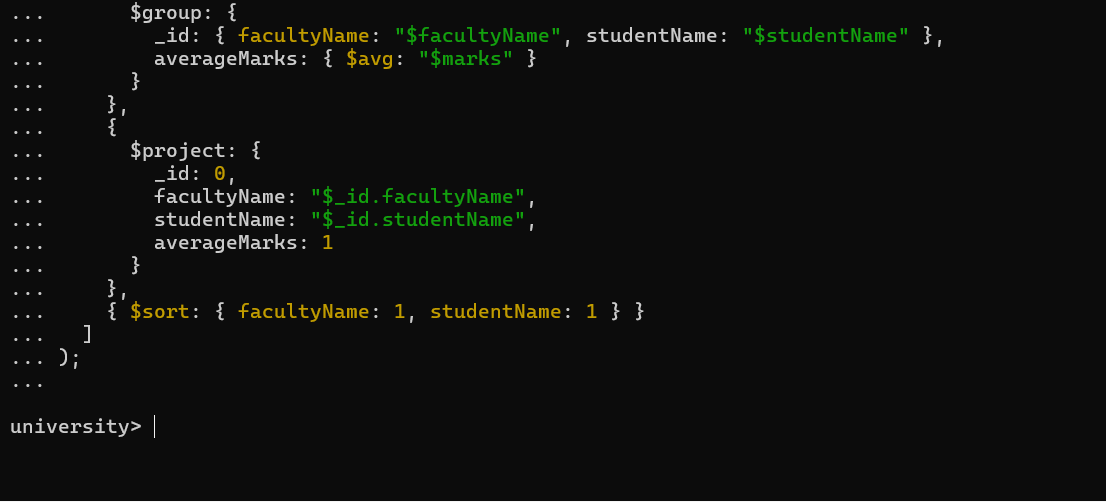
}

]

);

## Output:





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

## Solution:

db.students.aggregate( [

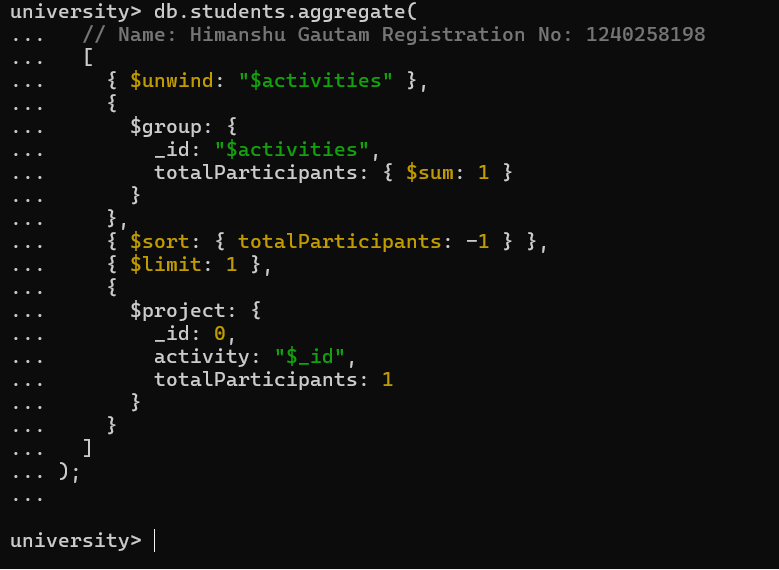
{ $unwind: "$activities" },

{

$group: {

\_id: "$activities", totalParticipants: { $sum: 1 }

## Output:

}

},

{ $sort: { totalParticipants: -1 }

},

{ $limit: 1 },

{

$project: {

\_id: 0,

activity: "$\_id", totalParticipants: 1

}

}

]

);