**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,

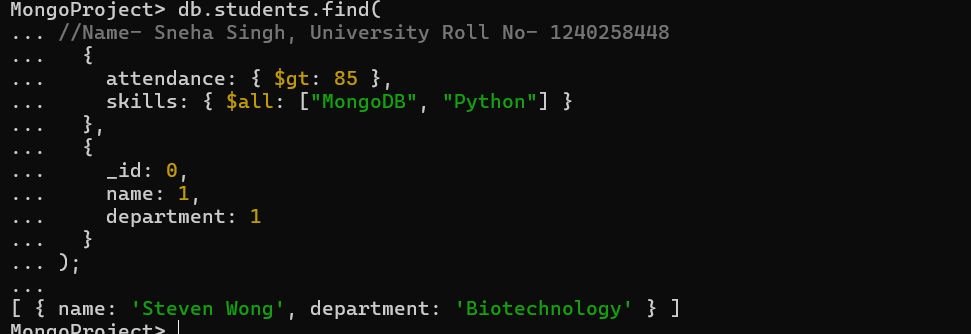
name: 1,

department: 1

}

);

Output:-



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: {

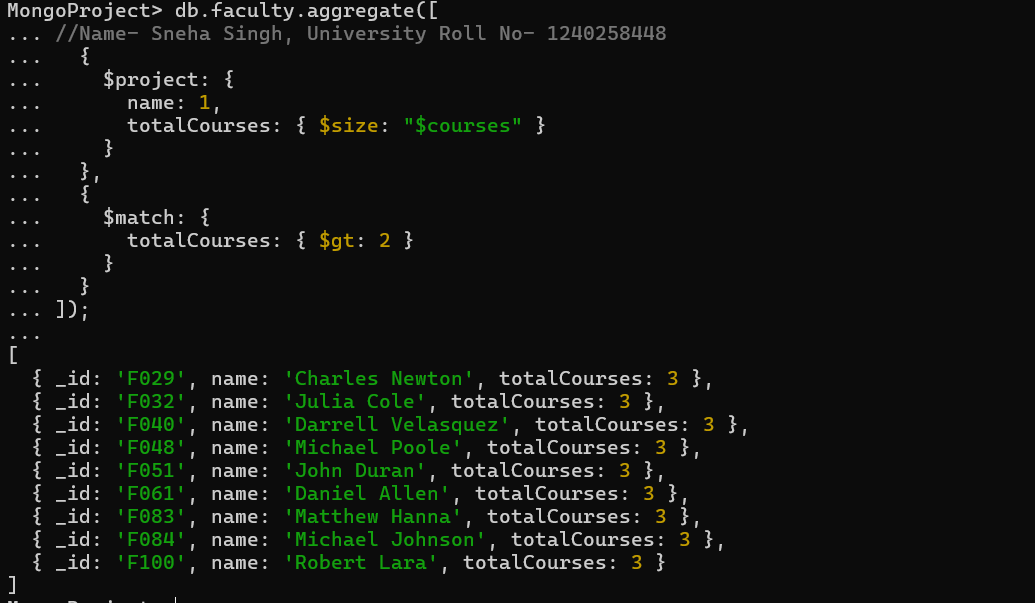
totalCourses: { $gt: 2 }

}

}

]);

Output:-



1. **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" },

{

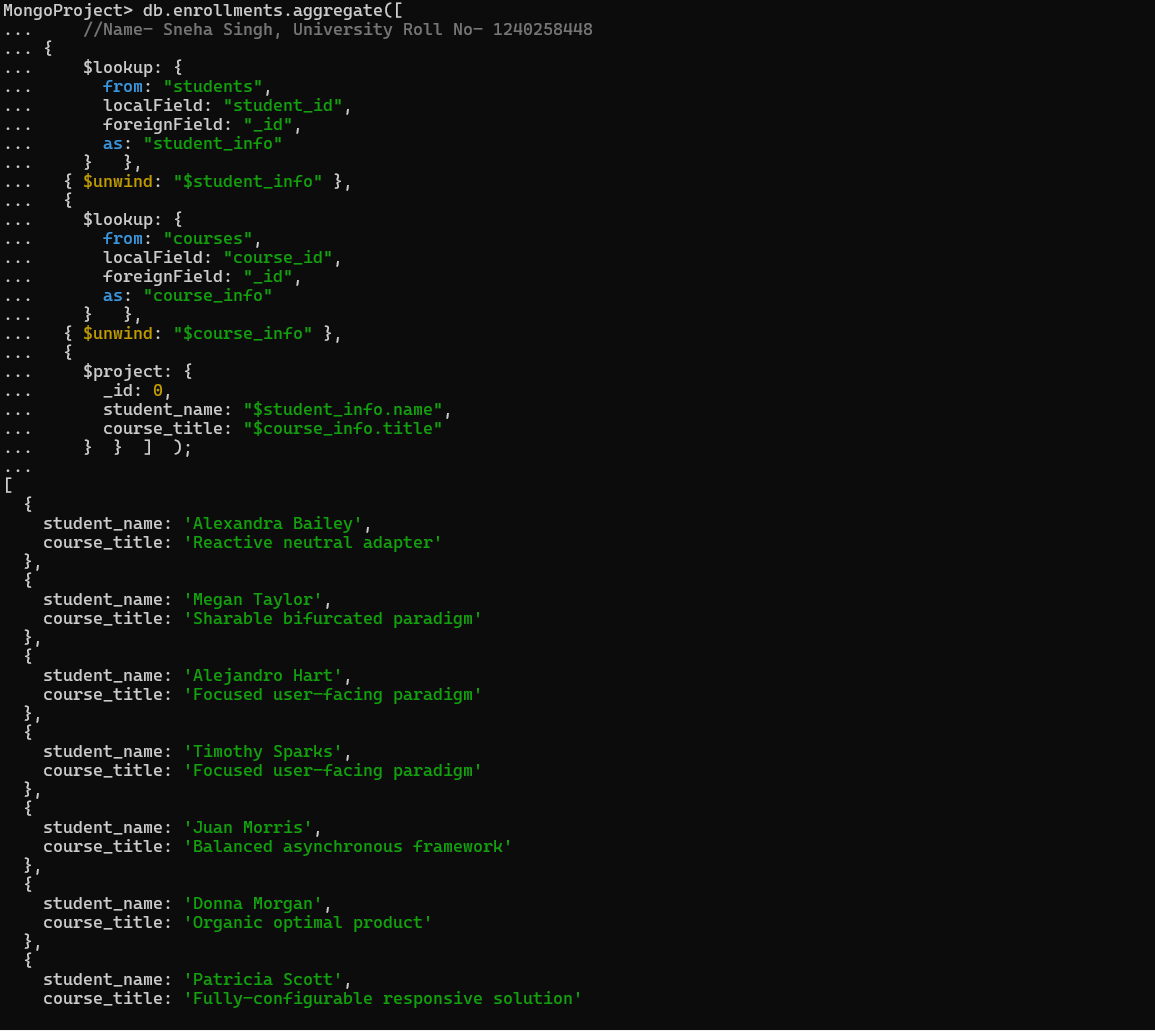
$project: {

\_id: 0,

student\_name: "$student\_info.name",

course\_title: "$course\_info.title"

} } ] );



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] },

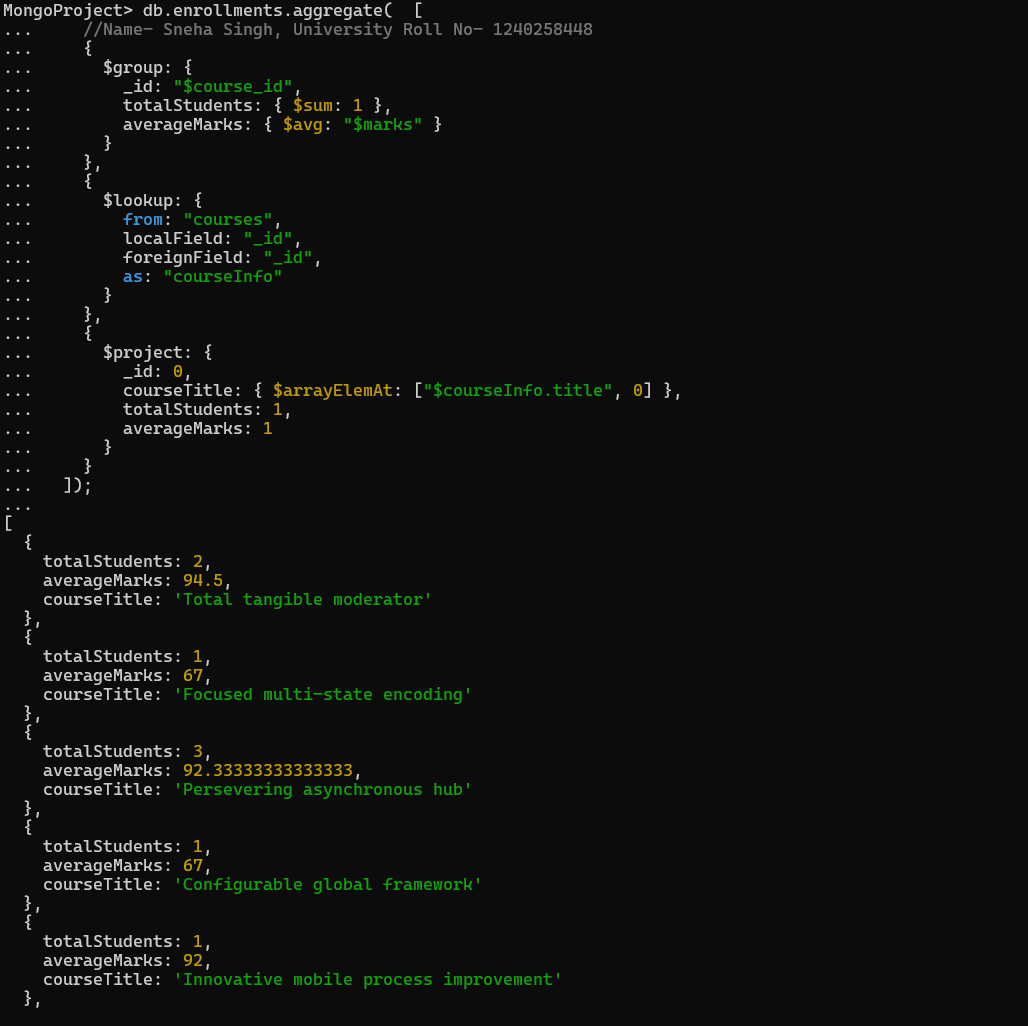
totalStudents: 1,

averageMarks: 1

}

}

]);



1. **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] }

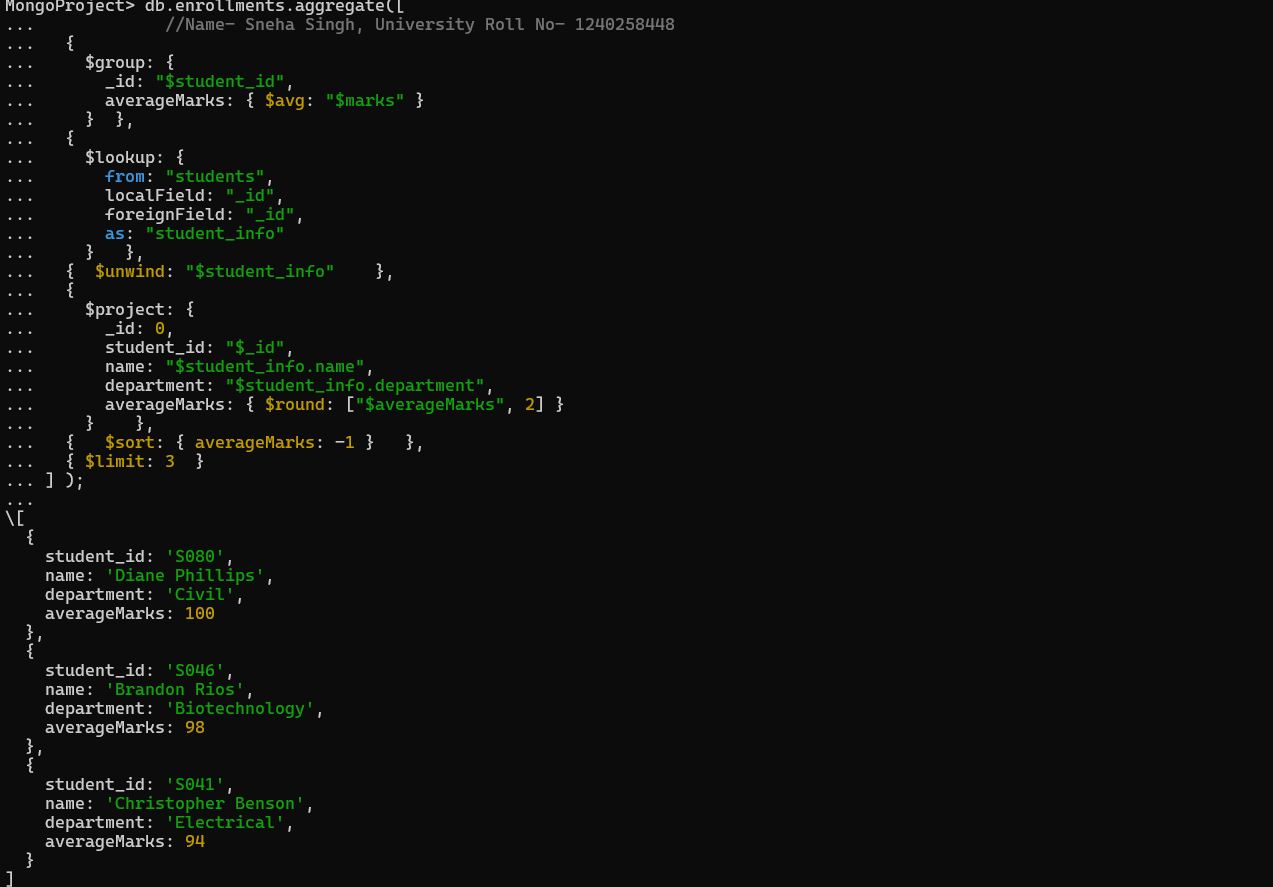
} },

{ $sort: { averageMarks: -1 } },

{ $limit: 3 }

] );

Output:-



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 }

}

},

{

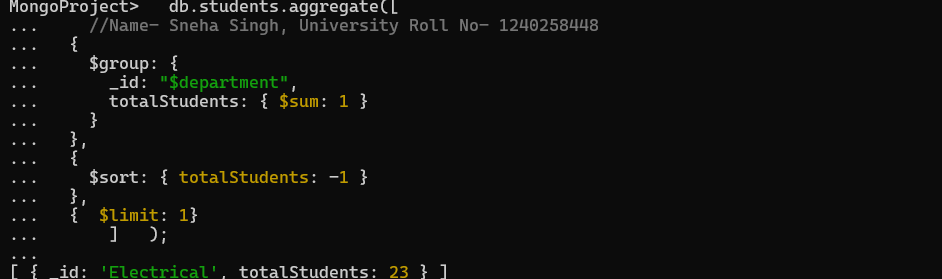
$sort: { totalStudents: -1 }

},

{ $limit: 1}

] );

Output:-



**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" })

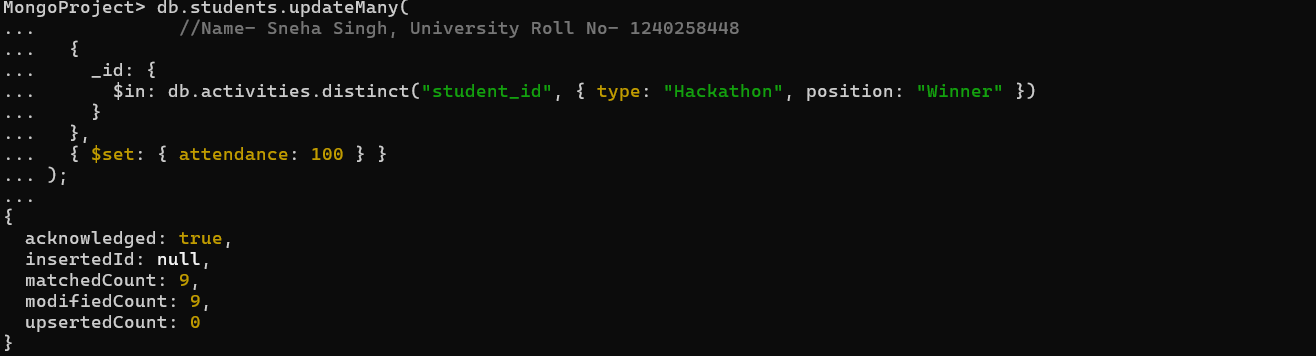
}

},

{ $set: { attendance: 100 } }

);

Output:-



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

Solution:

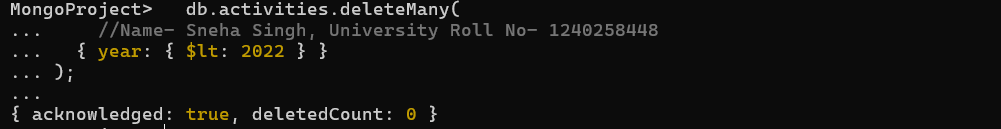
db.activities.deleteMany(

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

{ year: { $lt: 2022 } }

);

Output:-



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

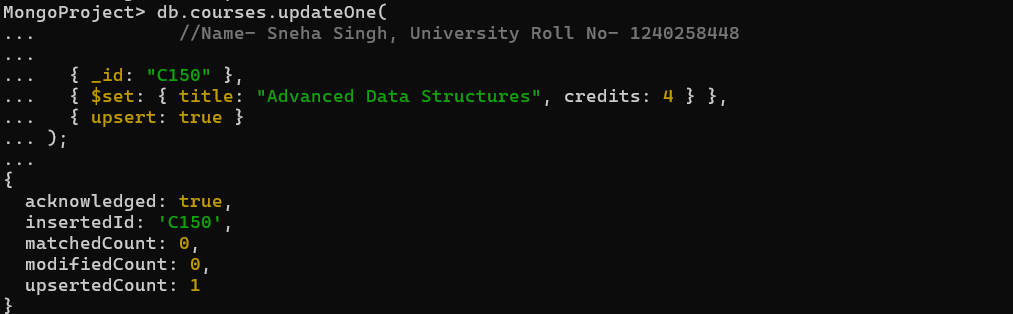
{ \_id: "C150" },

{ $set: { title: "Advanced Data Structures", credits: 4 } },

{ upsert: true }

);

Output:-



**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:-



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:-



**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:-



**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: {

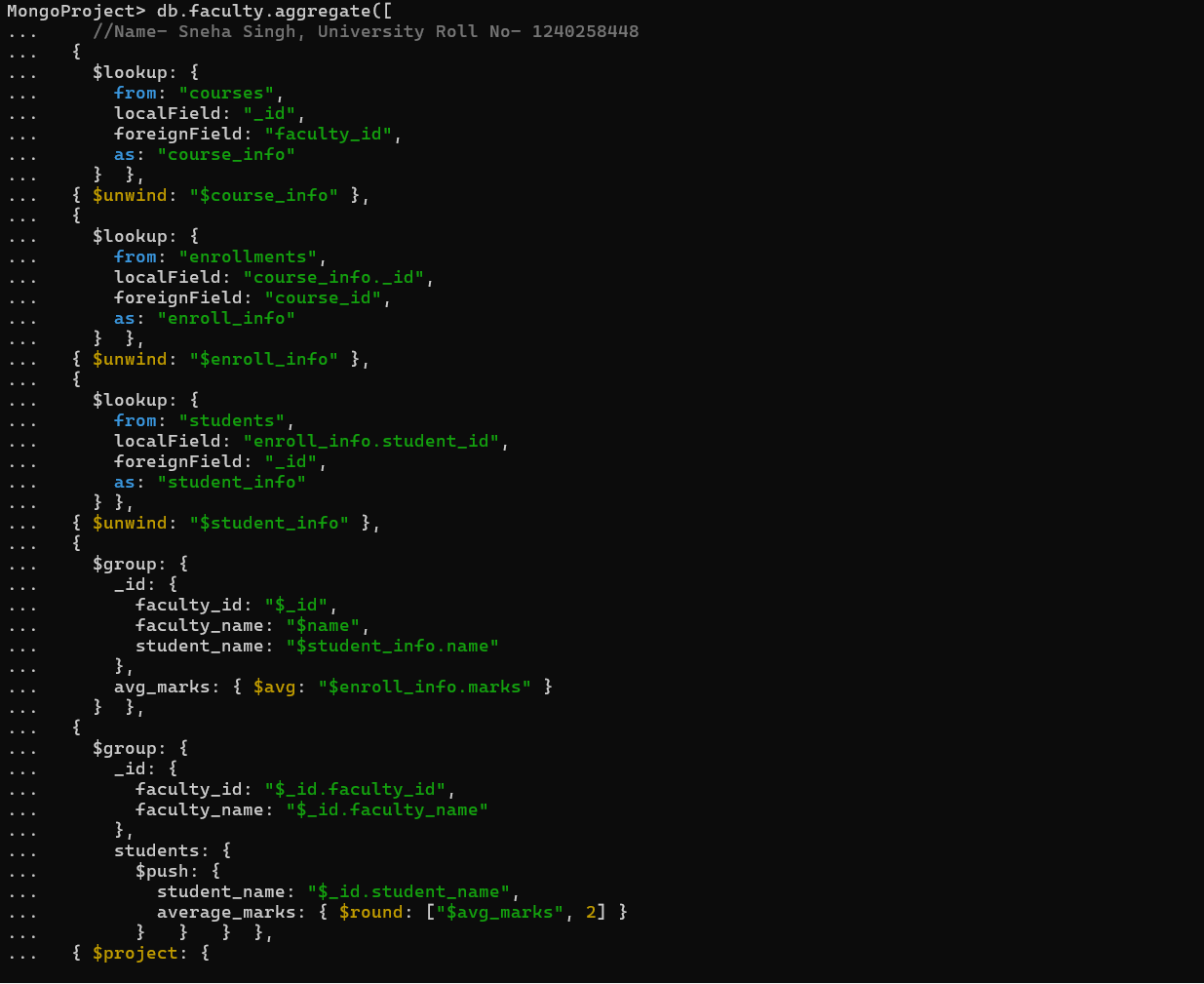
\_id: 0,

faculty\_name: "$\_id.faculty\_name",

students: 1 }

} ] );

Output:-





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:-

