Babu Banarasi Das University

**NAME: Diwas Pal**

**ROLL NO: 1240258175(32) SEC: BCADS23**

**SUB: NO SQL (Mongodb)**

**Submitted to : Mr. Ankit Soni**

**-:NO SQL MONGODB PROJECT:-**

Q1. List the names and departments of students who have more than 85% attendance and are skilled in both "MongoDB" and "Python".

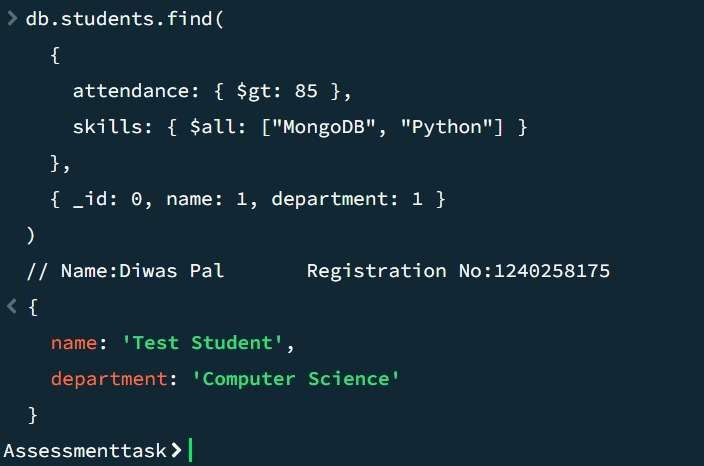
**ANS :**

**db.students.find(**

**{ attendance: { $gt: 85 }, skills: { $all: ["MongoDB", "Python"] } },**

**{ \_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

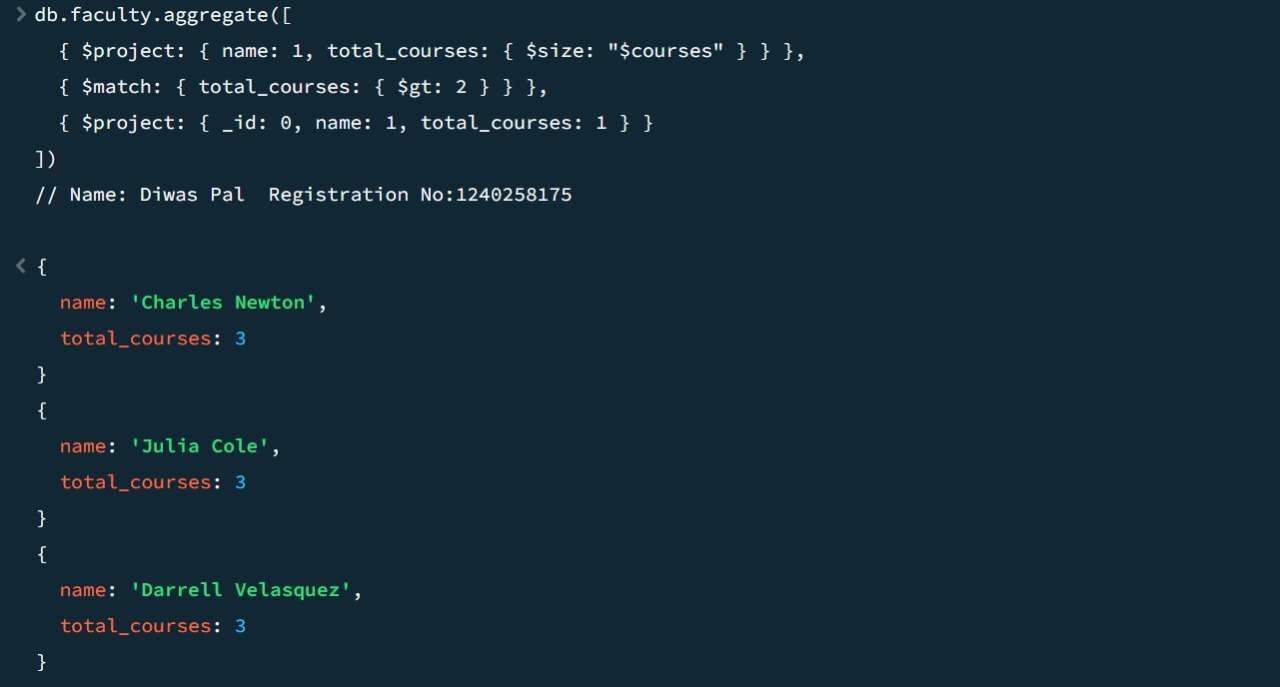
ANS:

db.faculty.aggregate([

{ $project: { \_id: 0, name: 1, totalCourses: { $size: "$courses" } } },

{ $match: { totalCourses: { $gt: 2 } } }

])



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).

db.students.aggregate([

{

$lookup: {

from: "enrollments",

localField: "\_id", foreignField: "student\_id", as: "enrolled\_courses"

}

},

{

$unwind: "$enrolled\_courses"

},

{

$lookup: {

from: "courses",

localField: "enrolled\_courses.course\_id", foreignField: "\_id",

as: "course\_details"

}

},

{

$unwind: "$course\_details"

},

{

$project: {

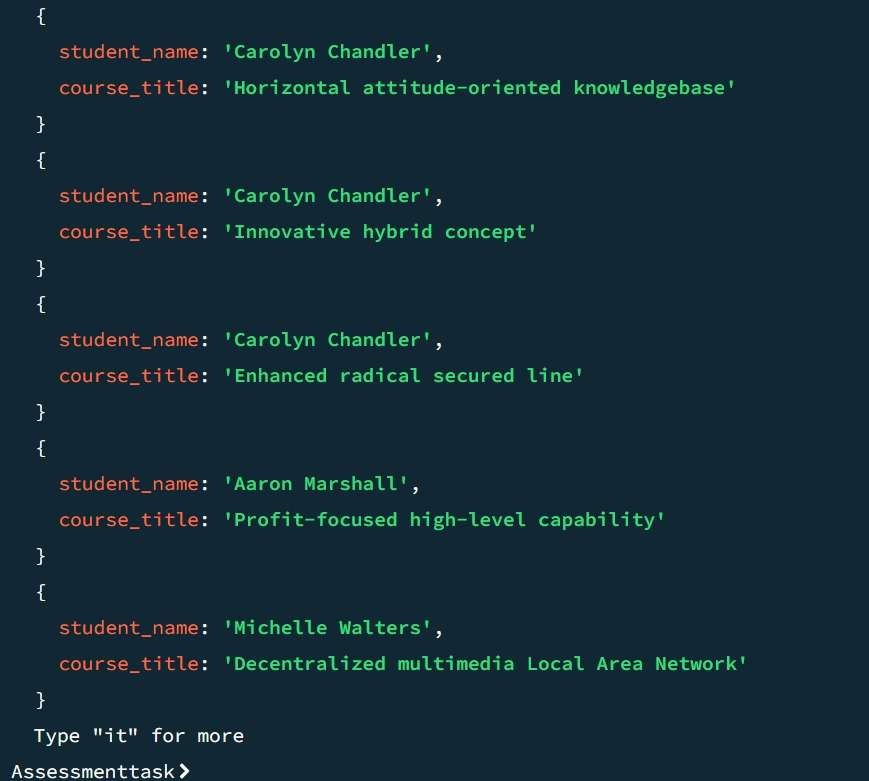
\_id: 0,

student\_name: "$name", course\_title: "$course\_details.title"

}

}

])



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

ANS:

db.enrollments.aggregate([

{

$lookup: {

from: "courses", localField: "course\_id", foreignField: "\_id",

as: "courseInfo"

}

},

{ $unwind: "$courseInfo" },

{

$group: {

\_id: "$courseInfo.title", totalStudents: { $sum: 1 }, avgMarks: { $avg: "$marks" }

}

},

{

$project: {

\_id: 0,

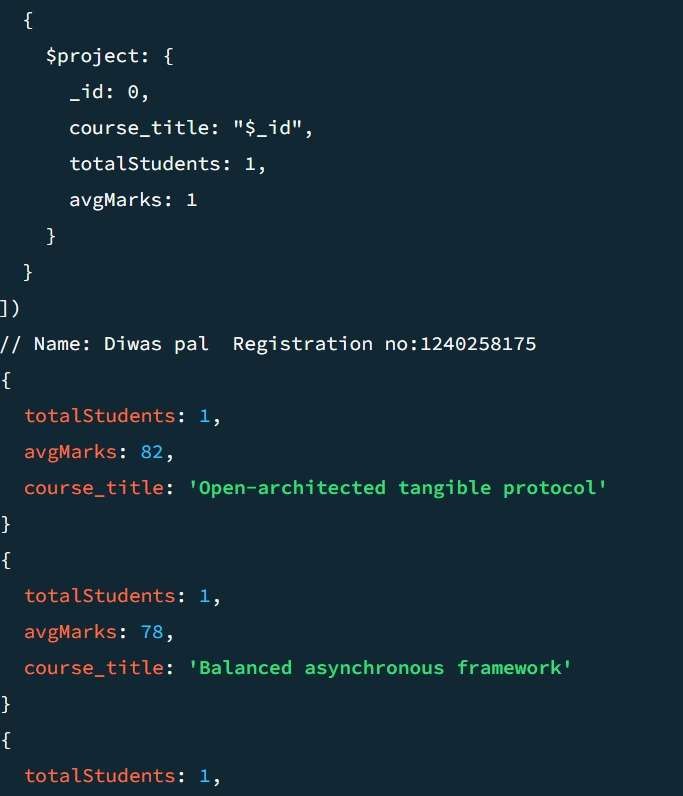
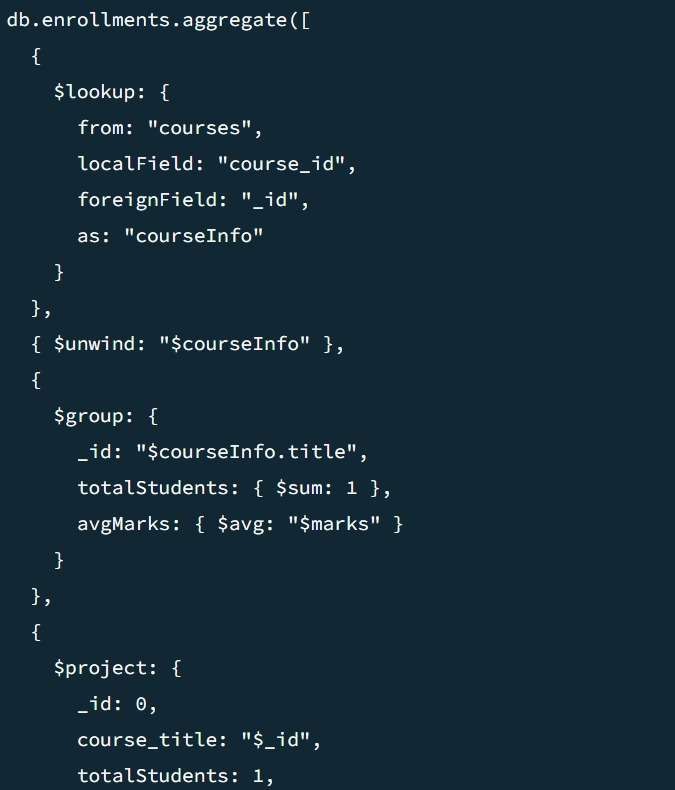
course\_title: "$\_id", totalStudents: 1,

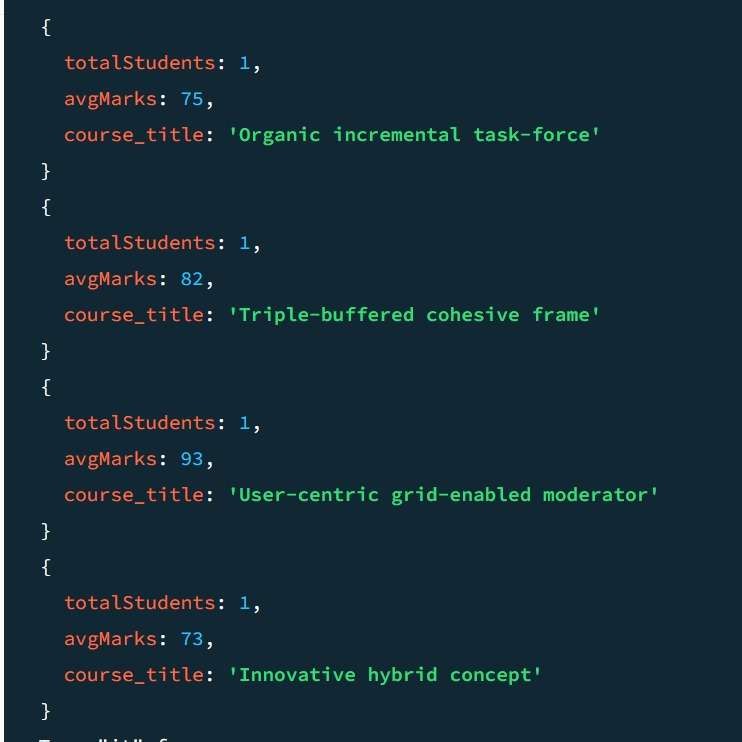
avgMarks: 1

}

}

])





Q5. Find the top 3 students with the highest average marks across all enrolled courses. db.enrollments.aggregate([

{

$group: {

\_id: "$student\_id", avgMarks: { $avg: "$marks" }

}

},

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

{ $limit: 3 },

{

$lookup: {

from: "students", localField: "\_id", foreignField: "\_id", as: "studentInfo"

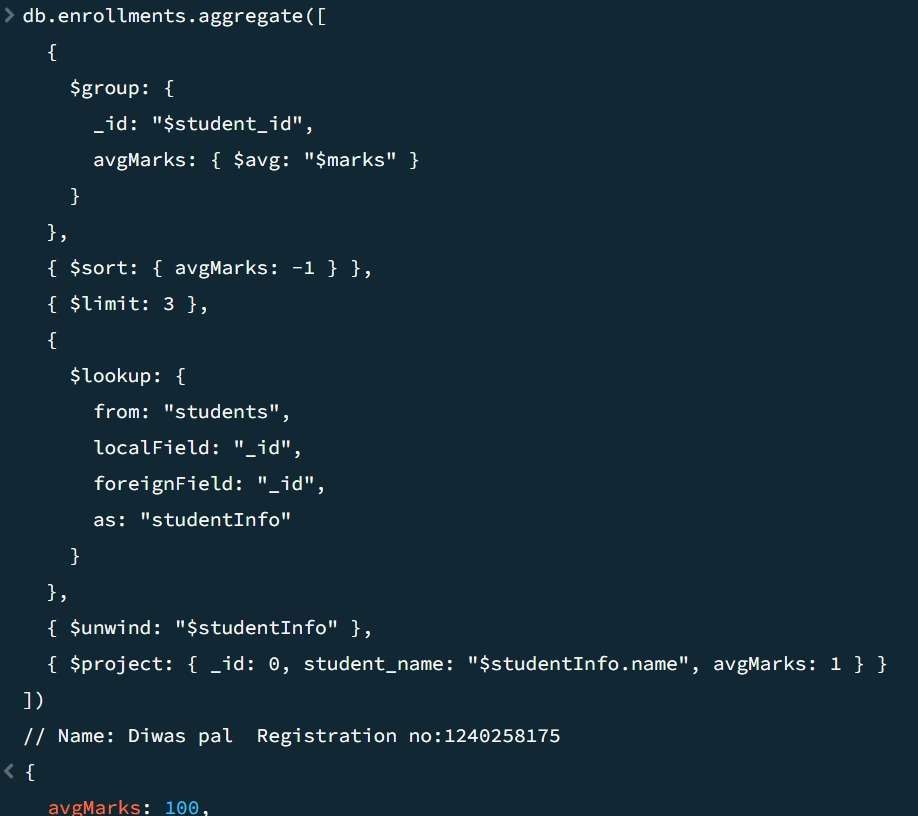
}

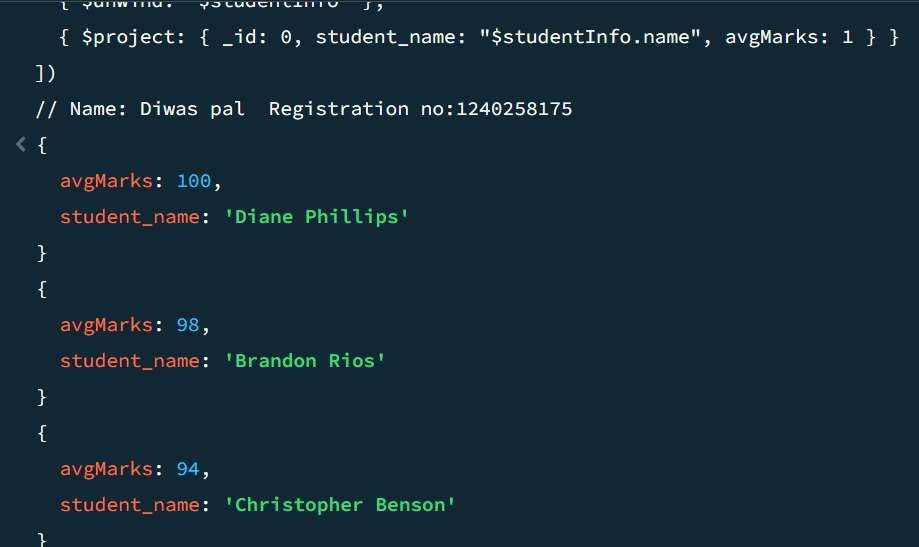
},

{ $unwind: "$studentInfo" },

{ $project: { \_id: 0, student\_name: "$studentInfo.name", avgMarks: 1 } }

])





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

db.students.aggregate([

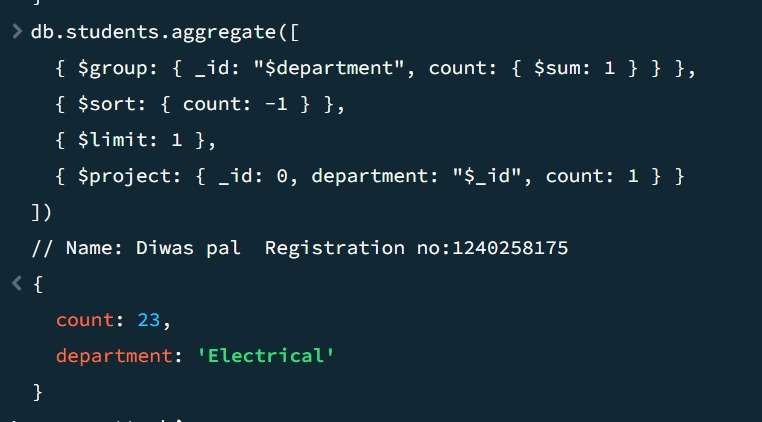
{ $group: { \_id: "$department", count: { $sum: 1 } } },

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

{ $limit: 1 },

{ $project: { \_id: 0, department: "$\_id", count: 1 } }

])



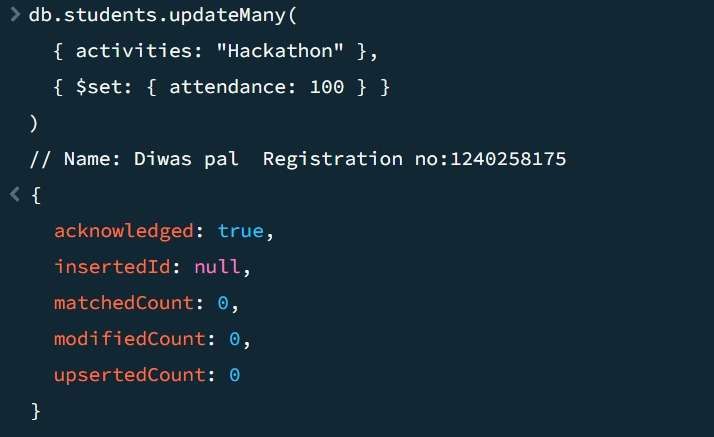
Q7. Update attendance to 100% for all students who won any "Hackathon".

db.students.updateMany(

{ activities: "Hackathon" },

{ $set: { attendance: 100 } }

)



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

db.student\_activities.deleteMany(

{ year: { $lt: 2022 } }

)



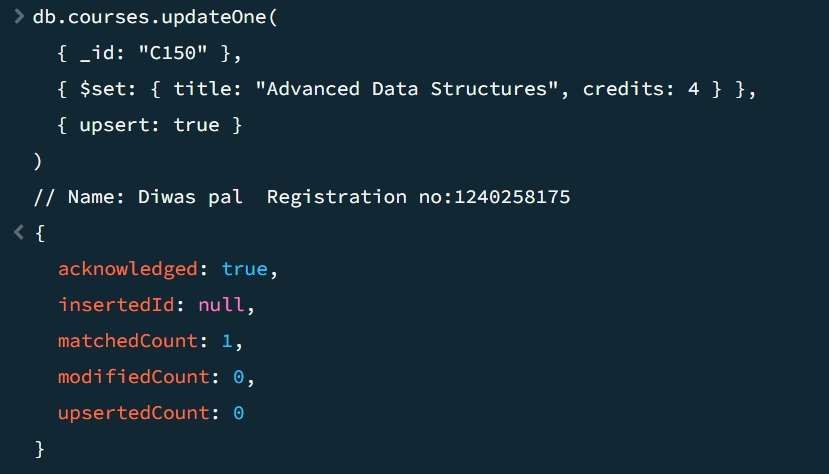
Q9. Upsert a course record for "Data Structures" with ID "C150" and credits 4—if it doesn’t exist, insert it; otherwise update its title to "Advanced Data Structures".

db.courses.updateOne(

{ \_id: "C150" },

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

{ upsert: true }

)

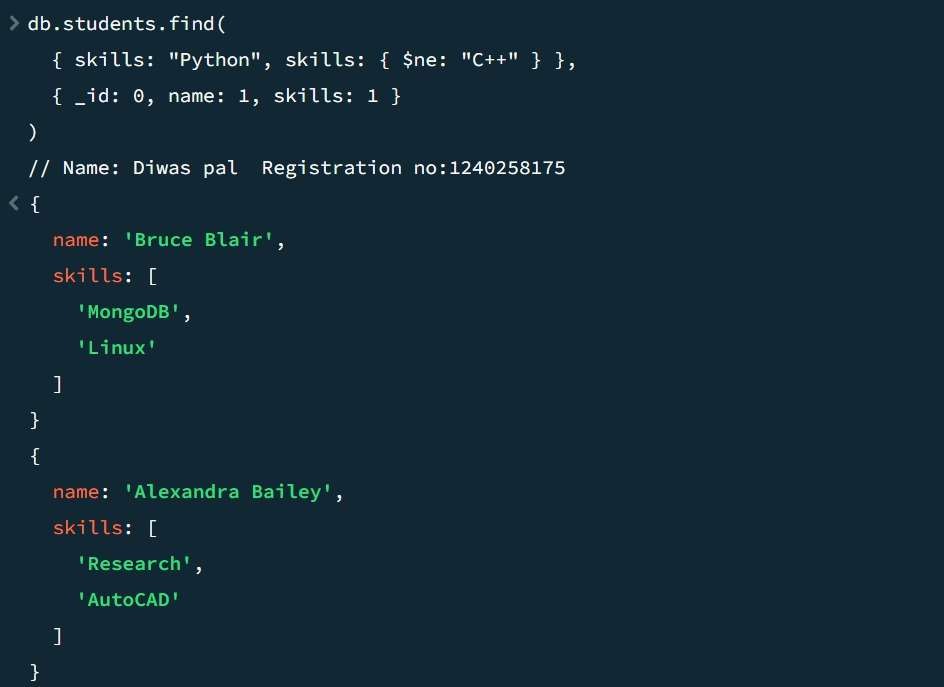
Q10. Find all students who have "Python" as a skill but not "C++".

db.students.find(

{ skills: "Python", skills: { $ne: "C++" } },

{ \_id: 0, name: 1, skills: 1 }

)



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

db.activities.aggregate([

{

$match: { type: { $in: ["Seminar", "Hackathon"] } }

},

{

$group: {

\_id: "$student\_id", name: { $first: "$name" },

types: { $addToSet: "$type" }

}

},

{

$match: { types: { $all: ["Seminar", "Hackathon"] } }

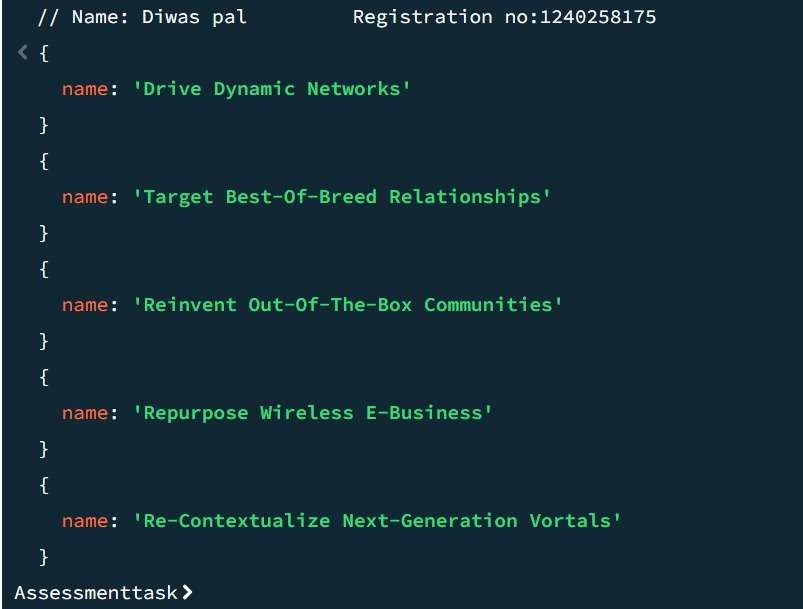
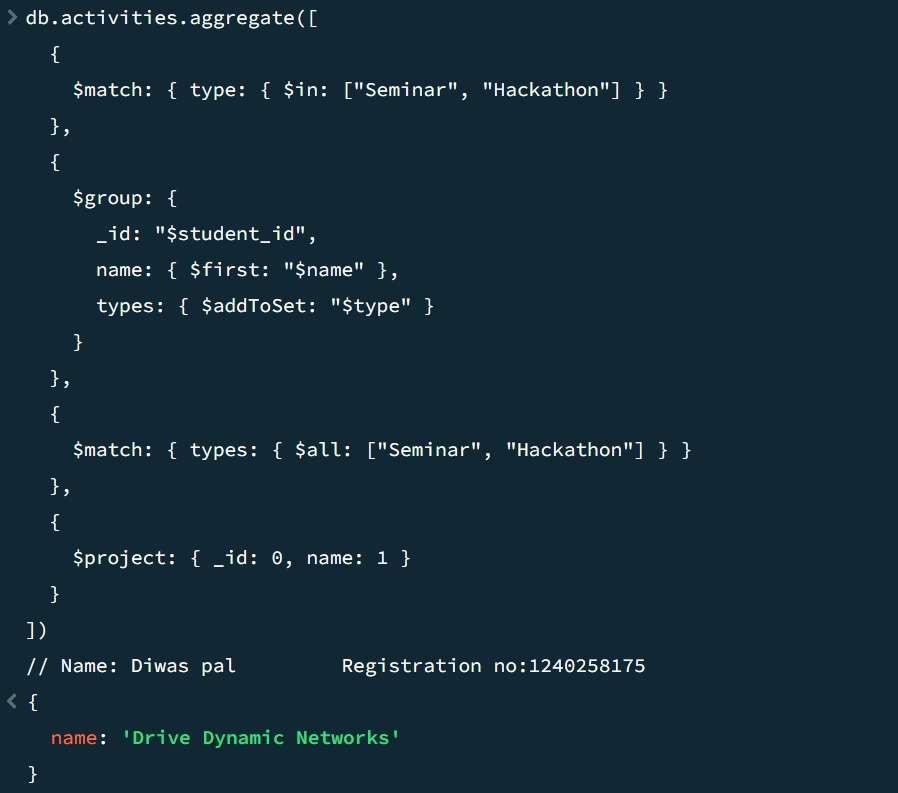
},

{

$project: { \_id: 0, name: 1 }

}

])



Q12. Find students who scored more than 80 in "Web Development" only if they belong to the "Computer Science" department.

db.students.find(

{

department: "Computer Science",

marks: { $elemMatch: { course: "Web Development", score: { $gt: 80 } } }

},

{ \_id: 0, name: 1, department: 1, "marks.$": 1 }

)



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

db.faculty.aggregate([

{ $unwind: "$courses" },

{

$lookup: {

from: "enrollments", localField: "courses", foreignField: "course\_id", as: "courseEnrollments"

}

},

{ $unwind: "$courseEnrollments" },

{

$lookup: {

from: "students",

localField: "courseEnrollments.student\_id", foreignField: "\_id",

as: "studentInfo"

}

},

{ $unwind: "$studentInfo" },

{

$group: {

\_id: { faculty: "$name", student: "$studentInfo.name" }, avgMarks: { $avg: "$courseEnrollments.marks" }

}

},

{

$group: {

\_id: "$\_id.faculty",

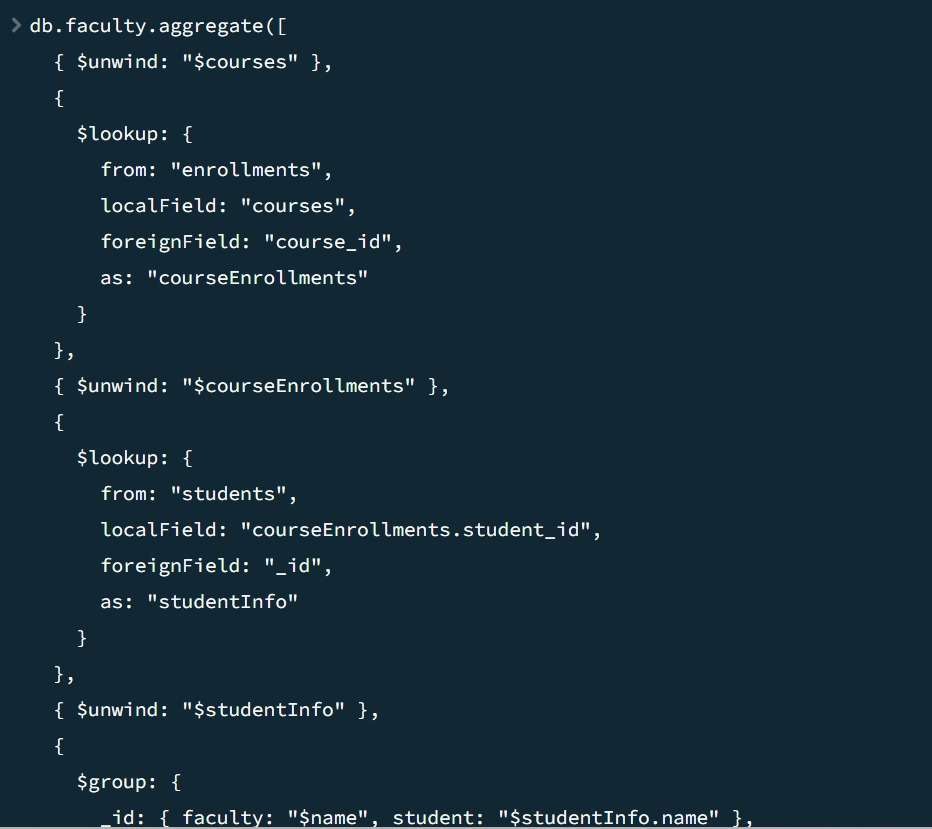
students: { $push: { student\_name: "$\_id.student", avgMarks: "$avgMarks" } }

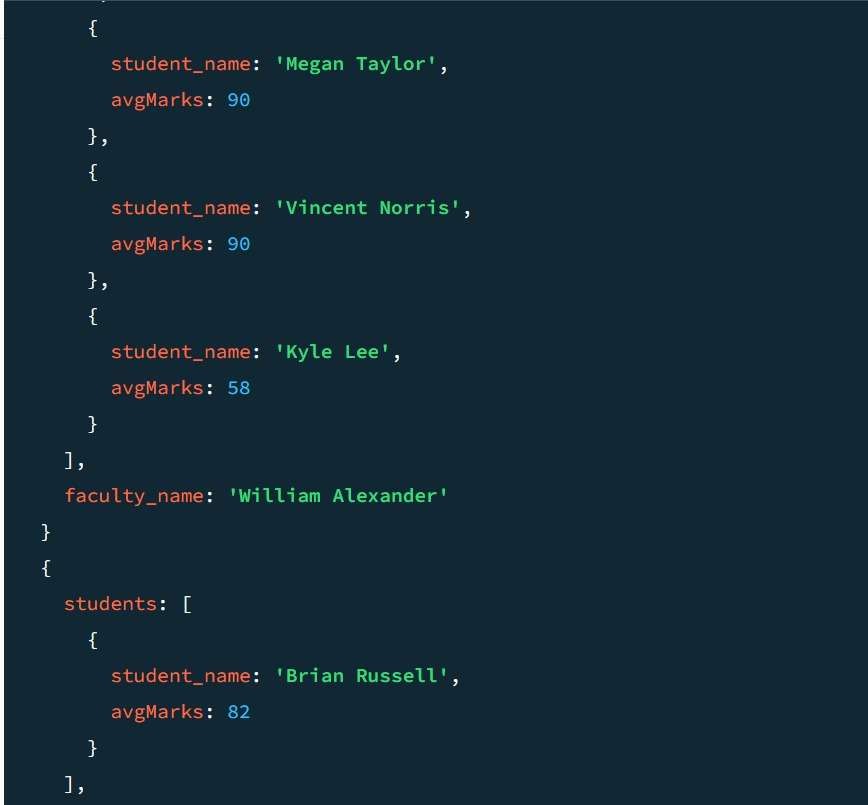
}

},

{ $project: { \_id: 0, faculty\_name: "$\_id", students: 1 } }

])





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

db.activities.aggregate([

{

$match: { type: { $in: ["Seminar", "Hackathon"] } }

},

{

$group: {

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

}

},

{

$project: { \_id: 0, type: "$\_id", totalParticipants: 1 }

}

])

