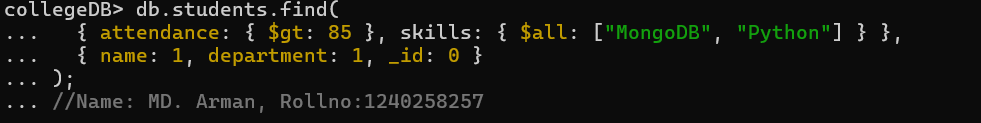
**Assessment Task 1**

# 1.Complex Filters & Projections

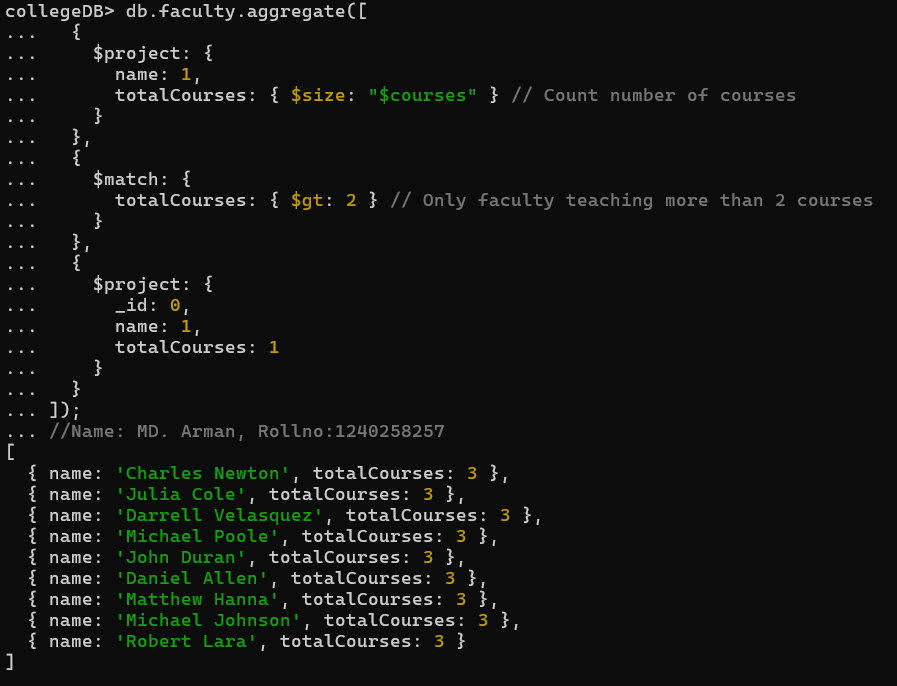
Q1. List the names and departments of students who have more than 85% attendance and are skilled in both "MongoDB" and "Python".

**Output:**

a

* find() → Retrieves documents from a collection based on conditions.
* $gt → “Greater than” comparison operator.
* $all → Matches arrays that contain all specified elements.

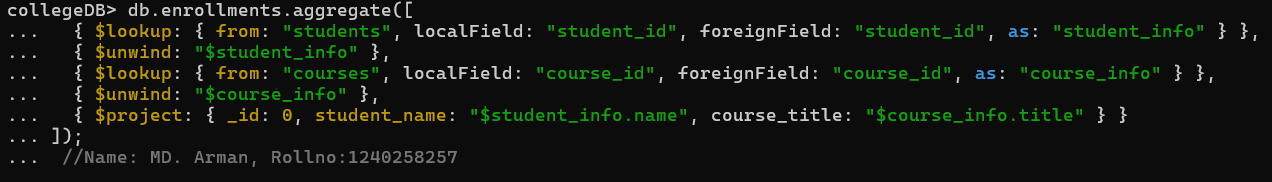
Q2. Show all faculty who are teaching more than 2 courses. Display their names and the total number of courses they teach.

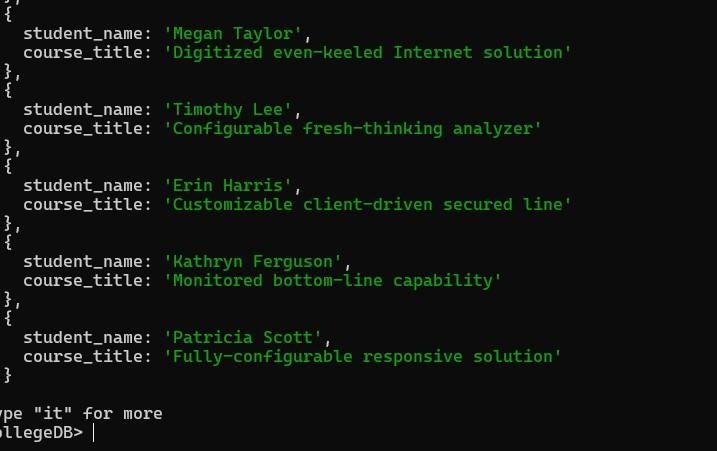
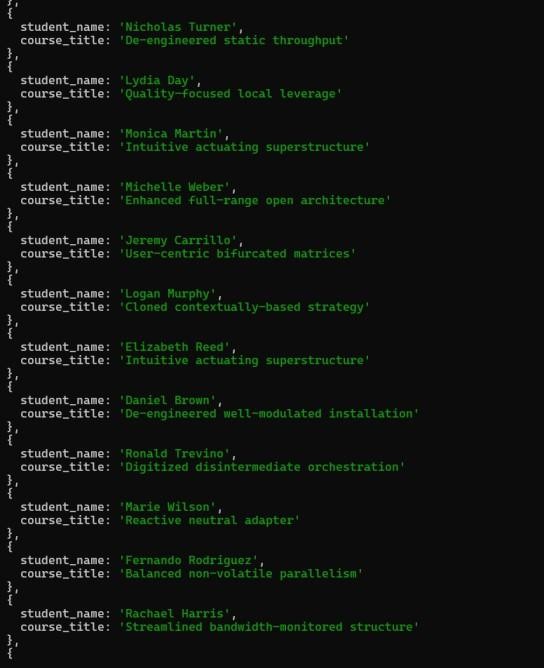


* $size: "$courses" → counts how many courses each faculty teaches
* Use $match to filter faculty members who are teaching more than 2 courses, using total Courses: { $gt: 2 }.
* Use $project to display only the fields needed in output: name and totalCourses, hide \_id.

## 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).

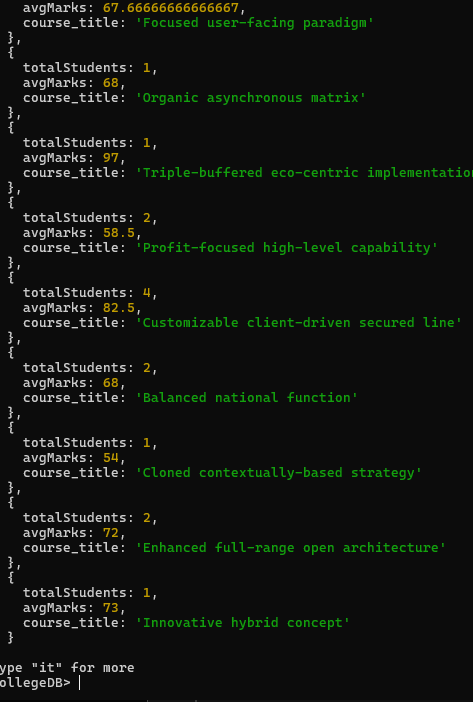
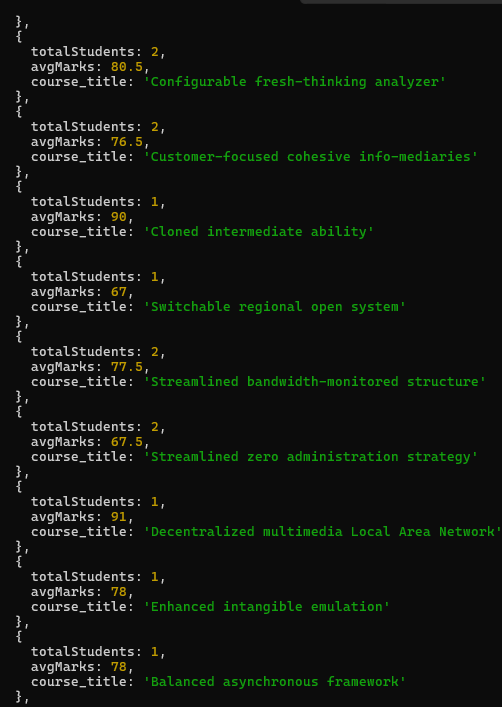




* Use $project to show only what we need:
* Student’s name → from student\_info.name •
* Course title → from course\_info.title
* Hide \_id so output looks clean.

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

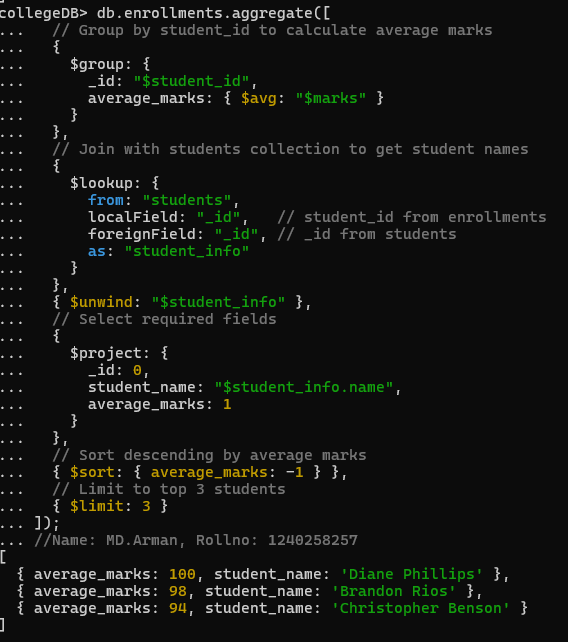




* totalStudents → counts how many students are enrolled using $sum: 1
* avgMarks → finds average marks using $avg: "$marks".
* Use $lookup to join this data with the courses collection.

# 3.Grouping, Sorting, and Limiting

Q5. Find the top 3 students with the highest average marks across all enrolled courses



$group → Groups documents by student\_id and calculates the average of marks.

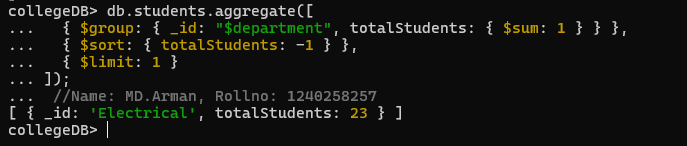
$lookup → Joins each grouped record with the students collection to get student details.

$unwind → Flattens the joined array from $lookup.

$project → Selects only student name and average marks.

$sort → Orders by average\_marks in descending order. $limit → Returns only the top 3 studen

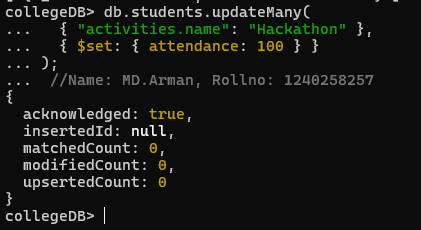
Q6. Count how many students are in each department. Display the department with the highest number of students.



* $group and $sum → Grouping and counting.
* $sort and $limit → Ranking results.

# 4.Update, Upsert, and Delete

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



* { "activities.name": "Hackathon" } → condition
* { $set: { attendance: 100 } } → sets attendance to 100%.

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

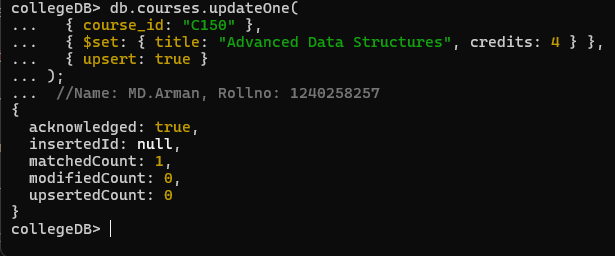


.

Deletes all activity records before 2022.

* deleteMany() → Removes multiple documents.
* $lt → “Less than” comparison operator.

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".



* { \_id: "C150" } → finds the course with ID "C150".
* Update Part: { $set: { title: "Advanced Data Structures", credits: 4 } }
* { upsert: true } : inserts if not found

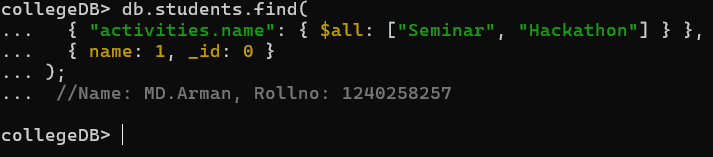
# 5. Array & Operator Usage

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



* $in → Matches documents containing Python.
* $nin → Matches documents that do **not** contain C++.

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



* Condition: activities.name: { $all: ["Seminar", "Hackathon"] }
* $all ensures the student has both "Seminar" and "Hackathon" in their activities array

# 6.Subdocuments and Nested Conditions

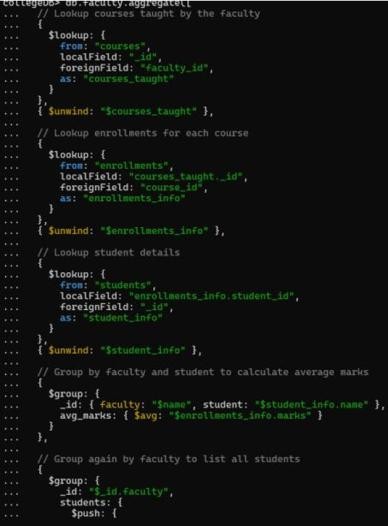
Q12. Find students who scored more than 80 in "Web Development" only if they belong to the "Computer Science" department.



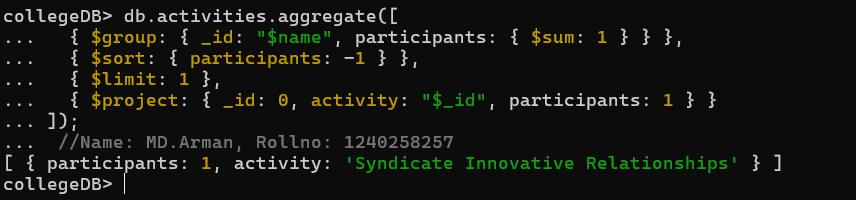
* Dot notation (marks.Web Development) → Accesses nested fields.
* $gt → Greater than condition.

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.



Q14. Show the most popular activity type (eg, Hackathon or Seminar) by number of student participants.



* $group and $sum → Count total participants.
* $sort and $limit → Rank and display the top result.
* $project → Rename and format final output.