

School of Computer Applications



No SQL and DbaaS 101 (BCADSN13202)

SUBMITTED TO:
Mr. Ankit Verma

SUBMITTED BY:
Name: Devansh Gupta
Roll No: 1240258159 (16)
Class: BCA DS&AI-23

MongoDB Assignment

Database Name: collegeDB

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

// Name: Devansh Gupta | Reg No: 1240258159

```
db.students.find(  
  {  
    attendance: { $gt: 85 },  
    skills: { $all: ["MongoDB", "Python"] }  
  },  
  { _id: 0, name: 1, department: 1 }  
)
```

> *// Name: Devansh Gupta | Reg No: 1240258159*

```
db.students.find(  
  {  
    attendance: { $gt: 85 },  
    skills: { $all: ["MongoDB", "Python"] }  
  },  
  { _id: 0, name: 1, department: 1 }  
)  
< {  
  name: 'Steven Wong',  
  department: 'Biotechnology'  
}
```

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

// Name: Devansh Gupta | Reg No: 1240258159

```
db.faculty.aggregate([
  { $project: { name: 1, totalCourses: { $size: "$courses" } } },
  { $match: { totalCourses: { $gt: 2 } } },
  { $project: { _id: 0, name: 1, totalCourses: 1 } }
])
```

```
< {
  studentName: 'Alexandra Bailey',
  courseTitle: 'Reactive neutral adapter'
}
{
  studentName: 'Megan Taylor',
  courseTitle: 'Sharable bifurcated paradigm'
}
{
  studentName: 'Alejandro Hart',
  courseTitle: 'Focused user-facing paradigm'
}
{
  studentName: 'Timothy Sparks',
  courseTitle: 'Focused user-facing paradigm'
}
{
  studentName: 'Juan Morris',
  courseTitle: 'Balanced asynchronous framework'
}
{
  studentName: 'Donna Morgan',
  courseTitle: 'Organic optimal product'
}
{
  studentName: 'Patricia Scott',
  courseTitle: 'Fully-configurable responsive solution'
}
{
```

```

{
  studentName: 'Chelsey Davis',
  courseTitle: 'Balanced national function'
}
{
  studentName: 'Timothy Sparks',
  courseTitle: 'Seamless high-level installation'
}
{
  studentName: 'Timothy Lee',
  courseTitle: 'Configurable fresh-thinking analyzer'
}
{
  studentName: 'Kathryn Ferguson',
  courseTitle: 'Monitored bottom-line capability'
}
{
  studentName: 'Lydia Day',
  courseTitle: 'Quality-focused local leverage'
}
{
  studentName: 'Nicholas Turner',
  courseTitle: 'De-engineered static throughput'
}
{
  studentName: 'Monica Martin',
  courseTitle: 'Intuitive actuating superstructure'
}
{
```

```

{
  studentName: 'Michelle Weber',
  courseTitle: 'Enhanced full-range open architecture'
}
{
  studentName: 'Logan Murphy',
  courseTitle: 'Cloned contextually-based strategy'
}
{
  studentName: 'Jeremy Carrillo',
  courseTitle: 'User-centric bifurcated matrices'
}
{
  studentName: 'Daniel Brown',
  courseTitle: 'De-engineered well-modulated installation'
}
{
  studentName: 'Elizabeth Reed',
  courseTitle: 'Intuitive actuating superstructure'
}

```

Q3. Show each student's name along with the course titles they are enrolled in.

// Name: Devansh Gupta | Reg No: 1240258159

```

db.enrollments.aggregate([
  {
    $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" },
  {
    $project: {
      _id: 0,
      studentName: "$student_info.name",
      courseTitle: "$course_info.title"
    }
  }
])

```

```

< {
  numStudents: 1,
  avgMarks: 74,
  courseTitle: 'Proactive optimizing initiative'
}
{
  numStudents: 1,
  avgMarks: 84,
  courseTitle: 'Seamless upward-trending project'
}
{
  numStudents: 1,
  avgMarks: 67,
  courseTitle: 'Configurable global framework'
}
{
  numStudents: 1,
  avgMarks: 86,
  courseTitle: 'Advanced analyzing budgetary management'
}
{
  numStudents: 1,
  avgMarks: 91,
  courseTitle: 'Decentralized multimedia Local Area Network'
}
{
  numStudents: 2,
  avgMarks: 77.5,
  courseTitle: 'Streamlined bandwidth-monitored structure'
}

```

MONOCOSH

```

}
{
  numStudents: 2,
  avgMarks: 58.5,
  courseTitle: 'Profit-focused high-level capability'
}
{
  numStudents: 2,
  avgMarks: 77.5,
  courseTitle: 'Configurable scalable data-warehouse'
}
{
  numStudents: 1,
  avgMarks: 93,
  courseTitle: 'User-centric grid-enabled moderator'
}
{
  numStudents: 2,
  avgMarks: 72,
  courseTitle: 'Enhanced full-range open architecture'
}
{
  numStudents: 2,
  avgMarks: 80.5,
  courseTitle: 'Configurable fresh-thinking analyzer'
}

```

```

{
  numStudents: 2,
  avgMarks: 77.5,
  courseTitle: 'Streamlined bandwidth-monitored structure'
}
{
  numStudents: 2,
  avgMarks: 58.5,
  courseTitle: 'Profit-focused high-level capability'
}
{
  numStudents: 2,
  avgMarks: 77.5,
  courseTitle: 'Configurable scalable data-warehouse'
}
{
  numStudents: 1,
  avgMarks: 93,
  courseTitle: 'User-centric grid-enabled moderator'
}
{
  numStudents: 2,
  avgMarks: 72,
  courseTitle: 'Enhanced full-range open architecture'
}
{
  numStudents: 2,
  avgMarks: 80.5,
  courseTitle: 'Configurable fresh-thinking analyzer'
}

```

```

{
  numStudents: 2,
  avgMarks: 80.5,
  courseTitle: 'Configurable fresh-thinking analyzer'
}
{
  numStudents: 1,
  avgMarks: 53,
  courseTitle: 'Fully-configurable reciprocal installation'
}
{
  numStudents: 1,
  avgMarks: 81,
  courseTitle: 'User-centric upward-trending functionalities'
}
{
  numStudents: 1,
  avgMarks: 50,
  courseTitle: 'Decentralized multi-tasking architecture'
}
{
  numStudents: 1,
  avgMarks: 90,
  courseTitle: 'Cloned intermediate ability'
}
{
  numStudents: 2,
  avgMarks: 67.5,
  courseTitle: 'Streamlined zero administration strategy'
}

```

```

{
  numStudents: 2,
  avgMarks: 67.5,
  courseTitle: 'Streamlined zero administration strategy'
}
{
  numStudents: 2,
  avgMarks: 64,
  courseTitle: 'Persistent static migration'
}
{
  numStudents: 2,
  avgMarks: 86.5,
  courseTitle: 'Fully-configurable responsive solution'
}
{
  numStudents: 1,
  avgMarks: 78,
  courseTitle: 'Balanced asynchronous framework'
}
{
  numStudents: 1,
  avgMarks: 82,
  courseTitle: 'Triple-buffered cohesive frame'
}

```

Q4. For each course, display the course title, number of students enrolled, and average marks.

// Name: Devansh Gupta | Reg No: 1240258159

```

db.enrollments.aggregate([
  {
    $group: {
      _id: "$course_id",
      numStudents: { $sum: 1 },
      avgMarks: { $avg: "$marks" }
    }
  },
  {
    $lookup: {
      from: "courses",
      localField: "_id",
      foreignField: "_id",
      as: "course_info"
    }
  },
  { $unwind: "$course_info" },
  {
    $project: {
      _id: 0,

```

```

        courseTitle: "$course_info.title",
        numStudents: 1,
        avgMarks: 1
    }
}
])

```

```

< {
  avgMarks: 100,
  studentName: 'Diane Phillips'
}
{
  avgMarks: 98,
  studentName: 'Brandon Rios'
}
{
  avgMarks: 94,
  studentName: 'Larry Ramsey'
}

```

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

// Name: Devansh Gupta | Reg No: 1240258159

```

db.enrollments.aggregate([
  { $group: { _id: "$student_id", avgMarks: { $avg: "$marks" } } },
  { $sort: { avgMarks: -1 } },
  { $limit: 3 },
  {
    $lookup: {
      from: "students",
      localField: "_id",
      foreignField: "_id",
      as: "student_info"
    }
  },
  { $unwind: "$student_info" },
  { $project: { _id: 0, studentName: "$student_info.name", avgMarks: 1 } }
])

```

```

> // Name: Devansh Gupta | Reg No: 1240258159

db.enrollments.aggregate([
  { $group: { _id: "$student_id", avgMarks: { $avg: "$marks" } } },
  { $sort: { avgMarks: -1 } },
  { $limit: 3 },
  {
    $lookup: {
      from: "students",
      localField: "_id",
      foreignField: "_id",
      as: "student_info"
    }
  },
  { $unwind: "$student_info" },
  { $project: { _id: 0, studentName: "$student_info.name", avgMarks: 1 } }
])
< {
  avgMarks: 100,
  studentName: 'Diane Phillips'
}
{
  avgMarks: 98,
  studentName: 'Brandon Rios'
}
{
  avgMarks: 94,
  studentName: 'Larry Ramsey'
}

```

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

// Name: Devansh Gupta | Reg No: 1240258159

```

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

```



```

> // Name: Devansh Gupta | Reg No: 1240258159

db.students.aggregate([
  { $group: { _id: "$department", totalStudents: { $sum: 1 } } },
  { $sort: { totalStudents: -1 } },
  { $limit: 1 }
])
< {
  _id: 'Electrical',
  totalStudents: 23
}

```

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

```

// Name: Devansh Gupta | Reg No: 1240258159
const winners = db.activities.distinct("student_id", { type: "Hackathon",
position: "Winner" })
db.students.updateMany(
  { _id: { $in: winners } },
  { $set: { attendance: 100 } }
)

```

```

> // Name: Devansh Gupta | Reg No: 1240258159

const winners = db.activities.distinct("student_id", { type: "Hackathon", position: "Winner" })
db.students.updateMany(
  { _id: { $in: winners } },
  { $set: { attendance: 100 } }
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 9,
  modifiedCount: 0,
  upsertedCount: 0
}

```

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

```

// Name: Devansh Gupta | Reg No: 1240258159

db.activities.deleteMany({ year: { $lt: 2022 } })

```

```

> // Name: Devansh Gupta | Reg No: 1240258159

db.activities.deleteMany({ year: { $lt: 2022 } })
< {
  acknowledged: true,
  deletedCount: 0
}

```

Q9. Upsert a course record for “Data Structures” with ID “C150” and credits 4.

// Name: Devansh Gupta | Reg No: 1240258159

```

db.courses.updateOne(
  { _id: "C150" },
  { $set: { title: "Advanced Data Structures", credits: 4 } },
  { upsert: true }
)

```

```

> // Name: Devansh Gupta | Reg No: 1240258159

db.courses.updateOne(
  { _id: "C150" },
  { $set: { title: "Advanced Data Structures", credits: 4 } },
  { upsert: true }
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}

```

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

// Name: Devansh Gupta | Reg No: 1240258159

```

db.students.find(
  {

```

```

    skills: { $in: ["Python"], $nin: ["C++"] }
  },
  { _id: 0, name: 1, skills: 1 }
)

```

```
> // Name: Devansh Gupta | Reg No: 1240258159
```

```

db.students.find(
  {
    skills: { $in: ["Python"], $nin: ["C++"] }
  },
  { _id: 0, name: 1, skills: 1 }
)
< {
  name: 'Kyle Hale',
  skills: [
    'Python',
    'Java'
  ]
}
{
  name: 'Cody Whitehead',
  skills: [
    'JavaScript',
    'Python'
  ]
}
{
  name: 'Thomas Jackson',
  skills: [
    'Python',
    'AutoCAD'
  ]
}

```

```

{
  name: 'Steven Wong',
  skills: [
    'MongoDB',
    'Python'
  ]
}
{
  name: 'Cheryl Jackson',
  skills: [
    'Research',
    'Python'
  ]
}
{
  name: 'Mr. Darius Newman',
  skills: [
    'Python',
    'SQL'
  ]
}
{
  name: 'Derrick Humphrey',
  skills: [
    'Python',
    'Java'
  ]
}
{

```

```
{
  name: 'Paula Jenkins',
  skills: [
    'JavaScript',
    'Python'
  ]
}
{
  name: 'Barbara Jones',
  skills: [
    'Python',
    'Research'
  ]
}
{
  name: 'Tracey Young',
  skills: [
    'Python',
    'AutoCAD'
  ]
}
{
  name: 'Elizabeth Reed',
  skills: [
    'Java',
    'Python'
  ]
}
{
```

```
{
  name: 'David Rivera',
  skills: [
    'Python',
    'JavaScript'
  ]
}
{
  name: 'Taylor Webb',
  skills: [
    'Linux',
    'Python'
  ]
}
{
  name: 'Erin Harris',
  skills: [
    'AutoCAD',
    'Python'
  ]
}
{
  name: 'Kyle Lee',
  skills: [
    'Python',
    'JavaScript'
  ]
}
}
```

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

// Name: Devansh Gupta | Reg No: 1240258159

```
const seminarStudents = db.activities.distinct("student_id", { type: "Seminar" })
const hackathonStudents = db.activities.distinct("student_id", { type: "Hackathon" })
const both = seminarStudents.filter(id => hackathonStudents.includes(id))
db.students.find({ _id: { $in: both } }, { _id: 0, name: 1 })
```

```
> // Name: Devansh Gupta | Reg No: 1240258159

const seminarStudents = db.activities.distinct("student_id", { type: "Seminar" })
const hackathonStudents = db.activities.distinct("student_id", { type: "Hackathon" })
const both = seminarStudents.filter(id => hackathonStudents.includes(id))
db.students.find({ _id: { $in: both } }, { _id: 0, name: 1 })
< {
  name: 'Adam Solomon'
}
{
  name: 'Lydia Day'
}
{
  name: 'Patricia Scott'
}
{
  name: 'Taylor Webb'
}
{
  name: 'Carlos Bryant'
}
```

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

// Name: Devansh Gupta | Reg No: 1240258159

```
const course = db.courses.findOne({ title: /Web Development/i })
if (course) {
  const topStudents = db.enrollments.distinct("student_id", { course_id: course._id, marks: { $gt: 80 } })
  db.students.find({ _id: { $in: topStudents }, department: "Computer Science" }, { _id: 0, name: 1, department: 1 })
}
```

```
// Name: Devansh Gupta | Reg No: 1240258159

const course = db.courses.findOne({ title: /Web Development/i })
if (course) {
  const topStudents = db.enrollments.distinct("student_id", { course_id: course._id, marks: { $gt: 80 } })
  db.students.find({ _id: { $in: topStudents }, department: "Computer Science" }, { _id: 0, name: 1, department: 1 })
}
```

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

// Name: Devansh Gupta | Reg No: 1240258159

```
db.courses.aggregate([
  {
    $lookup: {
      from: "enrollments",
      localField: "_id",
      foreignField: "course_id",
      as: "enrollments"
    }
  },
  { $unwind: "$enrollments" },
  {
    $lookup: {
      from: "students",
      localField: "enrollments.student_id",
      foreignField: "_id",
      as: "student_info"
    }
  },
  { $unwind: "$student_info" },
  {
    $group: {
      _id: { faculty_id: "$faculty_id", studentName: "$student_info.name" },
      avgMarks: { $avg: "$enrollments.marks" }
    }
  },
  {
    $lookup: {
      from: "faculty",
      localField: "_id.faculty_id",
      foreignField: "_id",
      as: "faculty_info"
    }
  },
  { $unwind: "$faculty_info" },
  {
    $project: {
      _id: 0,

```

```
        facultyName: "$faculty_info.name",
        studentName: "$_id.studentName",
        avgMarks: 1
    }
}
])
```

> // Name: Devansh Gupta | Reg No: 1240258159

```
db.courses.aggregate([
  {
    $lookup: {
      from: "enrollments",
      localField: "_id",
      foreignField: "course_id",
      as: "enrollments"
    }
  },
  { $unwind: "$enrollments" },
  {
    $lookup: {
      from: "students",
      localField: "enrollments.student_id",
      foreignField: "_id",
      as: "student_info"
    }
  },
  { $unwind: "$student_info" },
  {
    $group: {
      _id: { faculty_id: "$faculty_id", studentName: "$student_info.name" },
      avgMarks: { $avg: "$enrollments.marks" }
    }
  },
  {
    $lookup: {
      from: "faculty",
      localField: "_id.faculty_id",
      foreignField: "_id",
      as: "faculty_info"
    }
  },
  { $unwind: "$faculty_info" },
  {
    $project: {
      _id: 0,
      facultyName: "$faculty_info.name",
      studentName: "$_id.studentName",
      avgMarks: 1
    }
  }
])
< {
  avgMarks: 65,
  facultyName: 'April Palmer',
  studentName: 'Janet Jarvis'
}
{
  avgMarks: 56,
  facultyName: 'Jessica Campbell',
  studentName: 'Elizabeth Reed'
}
{
  avgMarks: 52,
  facultyName: 'Michele Hines',
  {
    avgMarks: 52,
    facultyName: 'Michele Hines',
    studentName: 'Joseph Brown'
  }
}
```

```
{
  avgMarks: 60,
  facultyName: 'Jessica Campbell',
  studentName: 'Bruce Blair'
}
{
  avgMarks: 85,
  facultyName: 'Kelly Huang',
  studentName: 'Thomas Jackson'
}
{
  avgMarks: 78,
  facultyName: 'James Martin',
  studentName: 'Jason Brown'
}
{
  avgMarks: 58,
  facultyName: 'Daniel Allen',
  studentName: 'Aaron Marshall'
}
{
  avgMarks: 91,
  facultyName: 'Julie Elliott',
  studentName: 'Michelle Walters'
}
{
  avgMarks: 64,
  facultyName: 'Kathryn Young',
  studentName: 'Jessica Galvan'
}
{
  avgMarks: 91,
  facultyName: 'Jacqueline Miller',
  studentName: 'Benjamin White'
}
{
  avgMarks: 51,
  facultyName: 'Michael Johnson',
  studentName: 'Carolyn Chandler'
}
{
  avgMarks: 84,
  facultyName: 'Cassandra Diaz',
  studentName: 'Ryan Flores'
}
{
  avgMarks: 79,
  facultyName: 'Kristina Young',
  studentName: 'Tina Hodge'
}
{
  avgMarks: 90,
  facultyName: 'Alexis Stone',
  studentName: 'Anthony Zavala'
}
{
  avgMarks: 84,
  facultyName: 'Robert Lara',
  studentName: 'Kathryn Ferguson'
}
```


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

// Name: Devansh Gupta | Reg No: 1240258159

```
db.activities.aggregate([
  { $group: { _id: "$type", totalParticipants: { $sum: 1 } } },
  { $sort: { totalParticipants: -1 } },
  { $limit: 1 }
])
```

> *// Name: Devansh Gupta | Reg No: 1240258159*

```
db.activities.aggregate([
  { $group: { _id: "$type", totalParticipants: { $sum: 1 } } },
  { $sort: { totalParticipants: -1 } },
  { $limit: 1 }
])
< {
  _id: 'Seminar',
  totalParticipants: 35
}
```
