

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 }  
)
```

```
> db.students.find(  
  {  
    attendance: { $gt: 85 },  
    skills: { $all: ["MongoDB", "Python"] }  
  },  
  { _id: 0, name: 1, department: 1 }  
)  
// Name:Diwas Pal      Registration No:1240258175  
< {  
  name: 'Test Student',  
  department: 'Computer Science'  
}
```

Assessmenttask> |

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 } } }
```

))

```
> db.faculty.aggregate([
  { $project: { name: 1, total_courses: { $size: "$courses" } } },
  { $match: { total_courses: { $gt: 2 } } },
  { $project: { _id: 0, name: 1, total_courses: 1 } }
])
// Name: Diwas Pal Registration No:1240258175

< {
  name: 'Charles Newton',
  total_courses: 3
}
{
  name: 'Julia Cole',
  total_courses: 3
}
{
  name: 'Darrell Velasquez',
  total_courses: 3
}
```

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"
  }
}
])
```

```

> 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"
    }
  }
])
// Name:Diwas Pal      Registration no:1240258175
< {
  student_name: 'Janet Jarvis',
  course_title: 'Configurable global framework'
}
{
  student_name: 'Janet Jarvis',
  course_title: 'Multi-lateral systemic success'
}
{
  student_name: 'Bruce Blair',
  course_title: 'Intuitive actuating superstructure'
}

```

```

{
  student_name: 'Carolyn Chandler',
  course_title: 'Horizontal attitude-oriented knowledgebase'
}
{
  student_name: 'Carolyn Chandler',
  course_title: 'Innovative hybrid concept'
}
{
  student_name: 'Carolyn Chandler',
  course_title: 'Enhanced radical secured line'
}
{
  student_name: 'Aaron Marshall',
  course_title: 'Profit-focused high-level capability'
}
{
  student_name: 'Michelle Walters',
  course_title: 'Decentralized multimedia Local Area Network'
}
Type "it" for more
Assessmenttask>

```

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
    }
  }
])
```

```

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,
    }
  }
])
// Name: Diwas pal Registration no:1240258175
{
  totalStudents: 1,
  avgMarks: 82,
  course_title: 'Open-architected tangible protocol'
}
{
  totalStudents: 1,
  avgMarks: 78,
  course_title: 'Balanced asynchronous framework'
}
{
  totalStudents: 1,

```

```

{
  totalStudents: 1,
  avgMarks: 75,
  course_title: 'Organic incremental task-force'
}
{
  totalStudents: 1,
  avgMarks: 82,
  course_title: 'Triple-buffered cohesive frame'
}
{
  totalStudents: 1,
  avgMarks: 93,
  course_title: 'User-centric grid-enabled moderator'
}
{
  totalStudents: 1,
  avgMarks: 73,
  course_title: 'Innovative hybrid concept'
}

```

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 } }
])
```



```

> 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 } }
])
// Name: Diwas pal  Registration no:1240258175
< {
  avgMarks: 100,

```

```

  { $unwind: "$studentInfo" },
  { $project: { _id: 0, student_name: "$studentInfo.name", avgMarks: 1 } }
])
// Name: Diwas pal  Registration no:1240258175
< {
  avgMarks: 100,
  student_name: 'Diane Phillips'
}
{
  avgMarks: 98,
  student_name: 'Brandon Rios'
}
{
  avgMarks: 94,
  student_name: 'Christopher Benson'
}

```

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 } }
])
```

```
> db.students.aggregate([
  { $group: { _id: "$department", count: { $sum: 1 } } },
  { $sort: { count: -1 } },
  { $limit: 1 },
  { $project: { _id: 0, department: "$_id", count: 1 } }
])
// Name: Diwas pal Registration no:1240258175
< {
  count: 23,
  department: 'Electrical'
}
```

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

```
db.students.updateMany(
  { activities: "Hackathon" },
  { $set: { attendance: 100 } }
)
```

```
> db.students.updateMany(
  { activities: "Hackathon" },
  { $set: { attendance: 100 } }
)
// Name: Diwas pal  Registration no:1240258175
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

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

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

```
> db.student_activities.deleteMany(
  { year: { $lt: 2022 } }
)
// Name: Diwas pal  Registration no:1240258175
< {
  acknowledged: true,
  deletedCount: 0
}
```

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 }  
)
```

```
> db.courses.updateOne(  
  { _id: "C150" },  
  { $set: { title: "Advanced Data Structures", credits: 4 } },  
  { upsert: true }  
)  
// Name: Diwas pal  Registration no:1240258175  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 0,  
  upsertedCount: 0  
}
```

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 }  
)
```

```
> db.students.find(
  { skills: "Python", skills: { $ne: "C++" } },
  { _id: 0, name: 1, skills: 1 }
)
// Name: Diwas pal  Registration no:1240258175
< {
  name: 'Bruce Blair',
  skills: [
    'MongoDB',
    'Linux'
  ]
}
{
  name: 'Alexandra Bailey',
  skills: [
    'Research',
    'AutoCAD'
  ]
}
```

```

{
  name: 'Kyle Hale',
  skills: [
    'Python',
    'Java'
  ]
}
{
  name: 'Daniel Robinson',
  skills: [
    'JavaScript',
    'Java'
  ]
}
{
  name: 'Tina Hodge',
  skills: [
    'SQL',
    'Research'
  ]
}
```

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 }
  }
])
```

```

> 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 }
  }
])
// Name: Diwas pal      Registration no:1240258175
< {
  name: 'Drive Dynamic Networks'
}

```

```

// Name: Diwas pal      Registration no:1240258175
< {
  name: 'Drive Dynamic Networks'
}
{
  name: 'Target Best-Of-Breed Relationships'
}
{
  name: 'Reinvent Out-Of-The-Box Communities'
}
{
  name: 'Repurpose Wireless E-Business'
}
{
  name: 'Re-Contextualize Next-Generation Vortals'
}
Assessmenttask>

```

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 }  
)
```



```

    },
    > db.students.find(
      {
        department: "Computer Science",
        attendance: { $gt: 80 }
      },
      { _id: 0, name: 1, department: 1, attendance: 1 }
    )// Name:Diwas Pal      Registration No:1240258175
< {
  name: 'Daniel Robinson',
  department: 'Computer Science',
  attendance: 92.1
}
{
  name: 'Mary Bennett',
  department: 'Computer Science',
  attendance: 83.36
}
{
  name: 'Ashley Myers',
  department: 'Computer Science',
  attendance: 95.26
}
}

```

```

{
  name: 'Christina Gordon',
  department: 'Computer Science',
  attendance: 86.27
}
{
  name: 'Taylor Webb',
  department: 'Computer Science',
  attendance: 82.91
}
{
  name: 'Carlos Bryant',
  department: 'Computer Science',
  attendance: 91.74
}
{
  name: 'Test Student',
  department: 'Computer Science',
  attendance: 90
}
}
Assessmenttask>|

```

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 } }
])

```

```

> 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" },

```

```

    {
      student_name: 'Megan Taylor',
      avgMarks: 90
    },
    {
      student_name: 'Vincent Norris',
      avgMarks: 90
    },
    {
      student_name: 'Kyle Lee',
      avgMarks: 58
    }
  ],
  faculty_name: 'William Alexander'
}
{
  students: [
    {
      student_name: 'Brian Russell',
      avgMarks: 82
    }
  ]
},

```

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 }
  }
])

```

```

> db.activities.aggregate([
  {
    $match: { type: { $in: ["Seminar", "Hackathon"] } }
  },
  {
    $group: {
      _id: "$type",
      totalParticipants: { $sum: 1 }
    }
  },
  {
    $project: { _id: 0, type: "$_id", totalParticipants: 1 }
  }
])
// Name: Diwas pal           Registration no:1240258175

< {
  totalParticipants: 35,
  type: 'Hackathon'
}
{
  totalParticipants: 35,
  type: 'Seminar'
}

```