

BABU BANARASI DAS UNIVERSITY
LUCKNOW
Session : 2025-2026



SCHOOL OF COMPUTER APPLICATIONS

PROJECT

No sql & mongo db

Submitted To:-
Mr. Ankit verma

Submitted By:-
Pal Deepanjali Radheshyam
Roll no:-1240258310

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

Output:

```
mongodb_project> db.students.find(
... //Name: Pal Deepanjali Radheshyam University Rollno.:1240258310
... {
...   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 the total number of courses they teach.

Solution:

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

Output:

```
mongodb_project> db.faculty.aggregate(  
... //Name: Pal Deepanjali Radheshyam University Rollno.: 1240258310  
... [ {  
... $project: {  
... name: 1,  
... totalCourses: { $size: "$courses" }  
... } },  
... {  
... $match: { totalCourses: { $gt: 2 } }  
... } ] );  
[  
  { _id: 'F029', name: 'Charles Newton', totalCourses: 3 },  
  { _id: 'F032', name: 'Julia Cole', totalCourses: 3 },  
  { _id: 'F040', name: 'Darrell Velasquez', totalCourses: 3 },  
  { _id: 'F048', name: 'Michael Poole', totalCourses: 3 },  
  { _id: 'F051', name: 'John Duran', totalCourses: 3 },  
  { _id: 'F061', name: 'Daniel Allen', totalCourses: 3 },  
  { _id: 'F083', name: 'Matthew Hanna', totalCourses: 3 },  
  { _id: 'F084', name: 'Michael Johnson', totalCourses: 3 },  
  { _id: 'F100', name: 'Robert Lara', totalCourses: 3 }  
]
```

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).

Solution:

```
db.enrollments.aggregate( [
  {
    $lookup: {
      from: "students",
      localField: "student_id",
      foreignField: "_id",
      as: "studentInfo"
    }
  },
  {
    $lookup: {
      from: "courses",
      localField: "course_id",
      foreignField: "_id",
      as: "courseInfo"
    }
  },
  {
    $project: {
      _id: 0,
      studentName: { $arrayElemAt: ["$studentInfo.name", 0] },
      courseTitles: "$courseInfo.title"
    }
  }
]
);
```

Output:

```
mongodb_project> db.enrollments.aggregate( [
... //Name: Pal Deepanjali Radheshyam University Rollno.: 1240258310
... {
... $lookup: {
... from: "students",
... localField: "student_id",
... foreignField: "_id",
... as: "studentInfo"
... },
... {
... $lookup: {
... from: "courses",
... localField: "course_id",
... foreignField: "_id",
... as: "courseInfo"
... },
... {
... $project: {
... _id: 0,
... studentName: { $arrayElemAt: ["$studentInfo.name", 0] },
... courseTitles: "$courseInfo.title"
... }
... }
... ];
[
  {
    studentName: 'Alexandra Bailey',
    courseTitles: [ 'Reactive neutral adapter' ]
  },
  {
    studentName: 'Megan Taylor',
    courseTitles: [ 'Sharable bifurcated paradigm' ]
  },
  {
    studentName: 'Alejandro Hart',
    courseTitles: [ 'Focused user-facing paradigm' ]
  },
  {
    studentName: 'Timothy Sparks',
    courseTitles: [ 'Focused user-facing paradigm' ]
  },
]
```

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

Solution:

```
db.enrollments.aggregate( [
  {
    $group: {
      _id: "$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
  }
}
]);

```

Output:

```

mongodb_project> db.enrollments.aggregate( [
... //Name: Pal Deepanjali Radheshyam University Rollno.: 1240258310
... {
...   $group: {
...     _id: "$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
...   }
... }
... ]]);
[
  {
    totalStudents: 3,
    averageMarks: 92.33333333333333,
    courseTitle: 'Persevering asynchronous hub'
  },
  {
    totalStudents: 2,
    averageMarks: 94.5,
    courseTitle: 'Total tangible moderator'
  },
  {
    totalStudents: 1,
    averageMarks: 82,
    courseTitle: 'Open-architected tangible protocol'
  },
]

```

3.Grouping, Sorting, and Limiting

Q.5 Find the top 3 students with the highest average marks across all enrolled courses

Solution:

```
db.enrollments.aggregate([
{
  $group: {
    _id: "$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:

```
mongodb_project> db.enrollments.aggregate([
... // Name: Pal Deepanjali Radheshyam University Rollno.: 1240258310
... {
...   $group: {
...     _id: "$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 }
... ]);
[
  {
    student_id: 'S080',
    name: 'Diane Phillips',
    department: 'Civil',
    averageMarks: 100
  },
  {
    student_id: 'S046',
    name: 'Brandon Rios',
    department: 'Biotechnology',
    averageMarks: 98
  },
  {
    student_id: 'S068',
    name: 'Larry Ramsey',
    department: 'Biotechnology',
    averageMarks: 94
  }
]
```

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

Solution:

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

```
},  
{ $limit: 1}  
] );
```

Output:

```
mongodb_project> db.students.aggregate([  
... // Name: Pal Deepanjali Radheshyam University Rollno.: 1240258310  
... {  
... $group: {  
... _id: "$department",  
... totalStudents: { $sum: 1 }  
... },  
... {  
... $sort: { totalStudents: -1 }  
... },  
... { $limit: 1 }  
... ] );  
[ { _id: 'Electrical', totalStudents: 23 } ]
```

4.Update, Upsert, and Delete.

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

Solution:

```
db.students.updateMany(  
{  
_id: {  
$in: db.activities.distinct("student_id", { type: "Hackathon", position: "Winner" })  
}  
},  
{ $set: { attendance: 100 } }  
);
```

Output:-

```
mongodb_project> db.students.updateMany(
... // Name: Pal Deepanjali Radheshyam University Rollno.: 1240258310
... {
...   _id: {
...     $in: db.activities.distinct("student_id", { type: "Hackathon", position: "Winner" })
...   },
...   { $set: { attendance: 100 } }
... }
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

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

Solution:

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

Output:

```
mongodb_project> db.activities_full.deleteMany(
... // Name: Pal Deepanjali Radheshyam University Rollno.: 1240258310
... { year: { $lt: 2022 } });
{ 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".

Solution:

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

Output:

```
mongodb_project> db.courses.updateOne(
... // Name: Pal Deepanjali Radheshyam University Rollno.: 1240258310
... { _id: "C150" },
... { $set: { title: "Advanced Data Structures", credits: 4 } },
... { upsert: true }
... );
{
  acknowledged: true,
  insertedId: 'C150',
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
```

5.Array & Operator Usage.

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

Solution:

```
db.students.find({
  $and: [
    { skills: "Python" },
    { skills: { $ne: "C++" } }
  ]
});
```

Output:

```

mongodb_project> db.students.find({
... // Name: Pal Deepanjali Radheshyam University Rollno.: 1240258310
... $and: [
... { skills: "Python" },
... { skills: { $ne: "C++" } }
... ]
... });
[
  {
    _id: 'S004',
    name: 'Kyle Hale',
    dob: '2000-10-20',
    department: 'Electrical',
    skills: [ 'Python', 'Java' ],
    attendance: 79.78
  },
  {
    _id: 'S008',
    name: 'Cody Whitehead',
    dob: '2003-11-25',
    department: 'Biotechnology',
    skills: [ 'JavaScript', 'Python' ],
    attendance: 92.03
  },
  {
    _id: 'S009',
    name: 'Thomas Jackson',
    dob: '2002-10-25',
    department: 'Electrical',
    skills: [ 'Python', 'AutoCAD' ],
    attendance: 96.64
  },
  {
    _id: 'S012',
    name: 'Steven Wong',
    dob: '2003-09-06',
    department: 'Biotechnology',
    skills: [ 'MongoDB', 'Python' ],
    attendance: 87.17
  },
]

```

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

Solution:

```

db.activities.aggregate([
{
  $group: {
    _id: "$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:

```

mongodb_project> db.activities.aggregate([
... // Name: Pal Deepanjali Radheshyam University Rollno.: 1240258310
... {
... $group: {
... _id: "$student_id",
... }},
... {
... $lookup: {
... from: "students",
... localField: "_id",
... foreignField: "_id",
... as: "student_details"
... }
... },
... {
... $project: {
... _id: 0,
... name: { $arrayElemAt: ["$student_details.name", 0] }
... } } ]);
[
{ name: 'Christina Gordon' },
{ name: 'Diane Phillips' },
{ name: 'Jeremy Carrillo' },
{ name: 'Dr. Michael Griffin Jr.' },
{ name: 'Kathryn Ferguson' },
{ name: 'Ronald Trevino' },
{ name: 'Monica Martin' },
{ name: 'Anthony Zavala' },
{ name: 'Daniel Brown' },
{ name: 'Donna Morgan' },
{ name: 'Jessica Galvan' },
{ name: 'Travis Johnson' },
{ name: 'Lydia Day' },

```

Q12. Find students who scored more than 80 in "Web Development" only if they belong to the "Computer Science" department. 7. Advanced Aggregation (Challenge Level).

Solution:

```
db.enrollments_full.aggregate([
  {$lookup: {
    from: "students_full",localField:"student_id",foreignField: "_id",as: "student"
  }},
  { $unwind: "$student" },
  {$lookup: {from: "courses_full",localField: "course_id",foreignField: "_id",as:
    "course"}},
  { $unwind: "$course" },
  {$match: {
    "marks": { $gt: 80 },
    "course.title": "Web
Development","student.department": "Computer Science"}},
  {$project: { _id: 0,student_name:
    "$student.name",
    course_title: "$course.title",marks: 1,
    department: "$student.department"}}
  ]);
```

Output:

```
PalDeepanjali> db.students_full.createIndex({ department: 1 }) // Pal Deepanjali Radheshyam -1240258310
department_1
PalDeepanjali> |
```

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([
{
  $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:

```

mongodb_project> db.faculty.aggregate([
... // Name: Pal Deepanjali Radheshyam University Rollno.: 1240258310
... {
...   $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 }
...             }
...           }
...         }
...       ]
...     }
...   ]
... ]

```

```

... }];
[
{
  students: [
    { student_name: 'Rachel Maldonado', average_marks: 71 },
    { student_name: 'Carolyn Chandler', average_marks: 51 },
    { student_name: 'Logan Murphy', average_marks: 54 }
  ],
  faculty_name: 'Michael Johnson'
},
{
  students: [
    { student_name: 'Steven Booth', average_marks: 69 },
    { student_name: 'Benjamin White', average_marks: 59 }
  ],
  faculty_name: 'Ann Porter MD'
},
{
  students: [
    { student_name: 'David Jones', average_marks: 67 },
    { student_name: 'Daniel Brown', average_marks: 75 }
  ],
  faculty_name: 'Kevin Murphy'
},
{
  students: [ { student_name: 'Jessica Galvan', average_marks: 64 } ],
  faculty_name: 'Kathryn Young'
},
{
  students: [
    { student_name: 'Elizabeth Reed', average_marks: 56 },
    { student_name: 'Monica Martin', average_marks: 92 },
    { student_name: 'Bruce Blair', average_marks: 60 }
  ],
  faculty_name: 'Jessica Campbell'
},
{
  students: [
    { student_name: 'Tracey Young', average_marks: 67 },
    { student_name: 'Ronald Trevino', average_marks: 93 }
  ],
  faculty_name: 'Maxwell Harrison'
},
{
  students: [
    { student_name: 'Timothy Sparks', average_marks: 80 },
    { student_name: 'Ryan Flores', average_marks: 84 }
  ],
  faculty_name: 'Cassandra Diaz'
},
{
  students: [
    { student_name: 'Anthony Zavala', average_marks: 90 },
    { student_name: 'Barbara Jones', average_marks: 93 }
  ],
  faculty_name: 'Alexis Stone'
},
]

```

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

Solution:

```
db.activities.aggregate([
  {
    $group: {
      _id: "$type",
      totalParticipants: { $addToSet: "$student_id" }
    } },
  {
    $project: {
      _id: 1,
      participantCount: { $size: "$totalParticipants" }
    } },
  {
    $sort: { participantCount: -1 }
  },
  {
    $limit: 1
  }
]);
```

```
PalDeepanjali> db.students_full.aggregate([
...   {
...     $group: {
...       _id: "$department",
...       avg_attendance: { $avg: "$attendance" }
...     }
...   },
...   { $sort: { avg_attendance: -1 } },
...   { $limit: 1 },
...   {
...     $project: {
...       _id: 0,
...       department: "$_id",
...       avg_attendance: 1
...     }
...   }
... ])
//Pal Deepanjali Radheshyam - 1240258310
[ { avg_attendance: 83.41526315789473, department: 'Biotechnology' } ]
PalDeepanjali> |
```