BABU BANARASI DAS UNIVERSITY

PROJECT
NOSQL & MongoDB
(BCADSN13202)

SUBMITTED TO :

Mr. Ankit Verma

SUBMITTED BY :

Name : Priya Maurya

Uni. Roll No : 1240258336

Section : BCADS25

Q .1 : List the names and departments of students who have more than 75% 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})

```
project_noSQL> db.students.find({attendance :  {$gt : 85} , skills : {$all : ["MongoDB" , "Python"]}} , {_id : 0 , name : 1, department : 1}) // PRIYA MAURY
A roll no 1240258336

project_noSQL>
```

Insights : Uses $all to ensure both skills exist.
Projection only returns name and department.

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

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

```
project_noSQL> db.faculty.aggregate([ { $project: { name: 1, number_of_courses: {$size: "$courses" } } }, { $match: { number_of_courses: {$gt: 2 } } } ]) /
 priya maurya 1240258336
[
  { _id: 'F029', name: 'Charles Newton', number_of_courses: 3 },
  { _id: 'F032', name: 'Julia Cole', number_of_courses: 3 },
  { _id: 'F040', name: 'Darrell Velasquez', number_of_courses: 3 },
  { _id: 'F048', name: 'Michael Poole', number_of_courses: 3 },
  { _id: 'F051', name: 'John Duran', number_of_courses: 3 },
  { _id: 'F061', name: 'Daniel Allen', number_of_courses: 3 },
  { _id: 'F083', name: 'Matthew Hanna', number_of_courses: 3 },
  { _id: 'F084', name: 'Michael Johnson', number_of_courses: 3 },
  { _id: 'F100', name: 'Robert Lara', number_of_courses: 3 }
]
```

Insights : $size counts array length.
$match filters faculty with > 2 courses.

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.enrollmets.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" },{$project: {_id: 0,student_name:"$student.name",course_title: "$course.title"}}])

```
project_noSQL> db.enrollmets.aggregate([{$lookup: {from:"students",localField: "student_id",foreignField: "_id",as: "student"}},{ $unwind:"$student" },{$loo
kup: {from: "courses",localField: "course_id",foreignField: "_id",as:"course"}},{ $unwind: "$course" },{$project: {_id: 0,student_name:"$student.name",cours
e_title: "$course.title"}}]) // priya maurya 1240258336
[
  {
    student_name: 'Alexandra Bailey',
    course_title: 'Reactive neutral adapter'
  },
  {
    student_name: 'Megan Taylor',
    course_title: 'Sharable bifurcated paradigm'
  },
  {
    student_name: 'Alejandro Hart',
    course_title: 'Focused user-facing paradigm'
  },
  {
    student_name: 'Timothy Sparks',
    course_title: 'Focused user-facing paradigm'
  },
  {
    student_name: 'Juan Morris',
    course_title: 'Balanced asynchronous framework'
```

Insights : Performs two joins ($lookup).

$unwind flattens joined arrays.

Q.4 : For each course, display the course title, number of students enrolled, and average marks (use $group).

Solution : db.enrollmets.aggregate([{ $lookup: { from: "courses", localField:"course_id", foreignField: "_id", as: "course" } }, { $unwind: "$course" }, { $group: { _id:"$course_id", course_title: { $first: "$course.title" }, total_students: { $sum: 1 }, average_marks: {$avg: "$marks" } } }])

```
project_noSQL> db.enrollmets.aggregate([ { $lookup: { from: "courses", localField:"course_id", foreignField: "_id", as: "course" } }, { $unwind: "$course" }
, { $group: { _id:"$course_id", course_title: { $first: "$course.title" }, total_students: { $sum: 1 }, average_marks: {$avg: "$marks" } } } ]) // priya
maurya 1240258336
[
  {
    _id: 'C046',
    course_title: 'Sharable responsive customer loyalty',
    total_students: 1,
    average_marks: 51
  },
  {
    _id: 'C085',
    course_title: 'Digitized disintermediate orchestration',
    total_students: 1,
    average_marks: 93
  },
  {
    _id: 'C094',
    course_title: 'Streamlined bandwidth-monitored structure',
    total_students: 2,
    average_marks: 77.5
  },
```

Insights : $group aggregates by course.

$avg computes average marks.

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

Solution : db.enrollmets.aggregate([{ $group : { _id : "$student_id" , average_marks : {$avg : "$marks"}}} , {$lookup : {from: "students" ,localField: "_id" , foreignField : "_id" , as : "student"} } , {$unwind : "$student" }, { $project : { _id : 0 , student_name : "$student.name" , average_marks : 1 } } , { $sort : { average_marks : -1}} , { $limit : 3}])

```
project_noSQL> db.enrollmets.aggregate([{ $group : { _id : "$student_id" , average_marks : {$avg : "$marks"}}} , {$lookup : {from: "students" ,localField: "
_id" , foreignField : "_id" , as : "student"} } , {$unwind : "$student" }, { $project : { _id : 0 , student_name : "$student.name" , average_marks : 1 } } ,
 { $sort : { average_marks : -1}} , { $limit : 3}]) // priya maurya 1240258336
[
  { average_marks: 100, student_name: 'Diane Phillips' },
  { average_marks: 98, student_name: 'Brandon Rios' },
  { average_marks: 94, student_name: 'Christopher Benson' }
]
```

Insights : Combines $group, $sort, $limit, $lookup.

Q.6 : Count how many students are in each department. Display the department with the highest number of students.

Solution : db.students.aggregate([{ $group: { _id: "$department", total_students: {$sum: 1 } } }, { $sort: { total_students: -1 } }, { $project: { _id: 0, department:"$_id", total_students: 1 } }])

```
project_noSQL> db.students.aggregate([ { $group: { _id: "$department", total_students: {$sum: 1 } } }, { $sort: { total_students: -1 } }, { $project: { _id:
 0, department:"$_id", total_students: 1 } } ]) // priya maurya 1240258336
[
  { total_students: 23, department: 'Electrical' },
  { total_students: 22, department: 'Civil' },
  { total_students: 20, department: 'Computer Science' },
  { total_students: 19, department: 'Biotechnology' },
  { total_students: 16, department: 'Mechanical' }
]
```

Insight : Combines $group, $sort, $project

Q .7 : Update attendance to 100% for all students who won any **"Hackathon"**.

Solution : db.students.updateMany({ activities : “Hackathon” }, { $set : { attendance : 100}})

```
project_noSQL> show dbs
BCA_NEW         16.00 KiB
admin           40.00 KiB
config          96.00 KiB
local           72.00 KiB
project_noSQL  212.00 KiB
project_noSQL> use project_noSQL
already on db project_noSQL
project_noSQL> show collections
Activities
courses
enrollmets
faculty
students
project_noSQL> db.students.updateMany({ activities : "Hackthon "} , { $set : { attendance : 100 }}) // PRIYA MAURYA roll no 1240258336
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

Insight : Use of $set for update data.

Q.8 : Delete all student activity records where the activity year is before 2022.

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

```
project_noSQL> db.Activities.deleteMany({ year : {$lt : 2022}}) // PRIYA MAURYA roll no 1240258336
{ acknowledged: true, deletedCount: 0 }
project_noSQL> db.Activities.find().limit(5)
[
  {
    _id: 'A001',
    student_id: 'S031',
    type: 'Hackathon',
    name: 'Streamline Frictionless Convergence',
    position: 'Participant',
    year: 2024
  },
  {
    _id: 'A002',
    student_id: 'S012',
    type: 'Hackathon',
    name: 'Drive World-Class Partnerships',
    position: 'Participant',
    year: 2024
  },
  {
    _id: 'A003',
    student_id: 'S019',
    type: 'Seminar',
    name: 'Mesh Transparent Deliverables',
    position: 'Winner',
    year: 2023
  },
```

Insight : Use of $lt with deleteMany.

Q.9 : Upsert a course record for **"Data Structures"** with ID "C156" and credits 4.

If it doesn't exist, insert it; otherwise, update its title to **"Advanced Data Structures".**

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

```
project_noSQL> db.courses.updateOne({ course_id : "C150" }, { $set : { title : "Advanced Data Structure", credits : 4}}, {upsert : true} ) //PRIYA MAURYA
roll no 1240258336
{
  acknowledged: true,
  insertedId: ObjectId('68efe1e2461c7dfc893b7f59'),
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
project_noSQL> db.courses.find().limit(3)
[
  {
    _id: 'C001',
    title: 'Profit-focused high-level capability',
    credits: 3,
    faculty_id: 'F061'
  },
  {
    _id: 'C002',
    title: 'Multi-lateral systemic success',
    credits: 2,
    faculty_id: 'F039'
  },
  {
    _id: 'C003',
    title: 'Reduced 5thgeneration complexity',
    credits: 3,
    faculty_id: 'F048'
  }
]
```

Insight : Combine use of $set , $upsert.

Q.10 : Find all students who have **"Python"** as a skill but not **"C++"**.

Solution : db.students.find({skills: {$in :["Python"], $nin: ["C++"] }},{ _id: 0,name: 1,skills:1})

```
project_noSQL> db.students.find({ skills : { $in : ["Python"] , $nin : ["C++"]}},{_id : 0, name : 1, skills :1}) //PRIYA MAURYA roll no 1240258336
[
  { 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: 'Brian Russell', skills: [ 'Python', 'Research' ] },
  { name: 'David Rivera', skills: [ 'Python', 'JavaScript' ] },
  { name: 'Taylor Webb', skills: [ 'Linux', 'Python' ] },
  { name: 'Erin Harris', skills: [ 'AutoCAD', 'Python' ] },
  { name: 'Kyle Lee', skills: [ 'Python', 'JavaScript' ] }
]
```

Insight : Use of $in , $nin to filter data .

Q.11 : Return names of students who participated in **"Seminar"** and **"Hackathon"** both.

Solution : db.Activities.find ({type : {$in : ["Seminar" , "Hackathon"]}} , {_id : 0 , name : 1 , type : 1})

```
project_noSQL> db.Activities.find({type : {$in : ["Seminar" , "Hackathon"]}} , {_id : 0 , name : 1 , type : 1})// PRIYA MAURYA roll no 1240258336
[
  { type: 'Hackathon', name: 'Streamline Frictionless Convergence' },
  { type: 'Hackathon', name: 'Drive World-Class Partnerships' },
  { type: 'Seminar', name: 'Mesh Transparent Deliverables' },
  { type: 'Seminar', name: 'Transition Global Communities' },
  { type: 'Seminar', name: 'Integrate Dot-Com Architectures' },
  { type: 'Hackathon', name: 'Architect Dot-Com Users' },
  { type: 'Seminar', name: 'Integrate B2C Functionalities' },
  { type: 'Seminar', name: 'Aggregate Distributed E-Tailers' },
  { type: 'Seminar', name: 'Monetize 24/7 Synergies' },
  { type: 'Seminar', name: 'Envisioneer Vertical Schemas' },
  { type: 'Hackathon', name: 'Benchmark Proactive Bandwidth' },
  { type: 'Hackathon', name: 'Expedite Customized Bandwidth' },
  { type: 'Hackathon', name: 'Scale Real-Time Eyeballs' },
  { type: 'Hackathon', name: 'Architect Collaborative Eyeballs' },
  { type: 'Hackathon', name: 'Re-Contextualize Real-Time Eyeballs' },
  { type: 'Hackathon', name: 'Reinvent Out-Of-The-Box Communities' },
  { type: 'Hackathon', name: 'Integrate Interactive Channels' },
  { type: 'Seminar', name: 'Innovate Ubiquitous Technologies' },
  { type: 'Seminar', name: 'Target Innovative Partnerships' },
  { type: 'Hackathon', name: 'Innovate Intuitive Info-Mediaries' }
]
```

Insight : Filter those students who participated in “Hackathon” , “Seminar”.

Q.12 : Find students whose attendance more than 95 % only if they belong to the **"Computer Science"** department.

Solution : db.students.find({ department : “Computer Science” , attendance : { $gt : 95}] , { _id : 0, department: 1 , attendance : 1})

```
project_noSQL> db.students.find({ department: "Computer Science", attendance: { $gt: 95 } }, { _id: 0, name: 1, department: 1, attendance: 1 }) // PRIYA
MAURYA roll no 1240258336
[
  {
    name: 'Ashley Myers',
    department: 'Computer Science',
    attendance: 95.26
  },
  {
    name: 'Elizabeth Reed',
    department: 'Computer Science',
    attendance: 95.53
  }
]
```

Insight : Filter students on attendance more than 95 % using $gt.

Q.13 : For each faculty member, list the names of all students enrolled in their courses along with average marks per student per faculty.

Solution : db.courses.aggregate([{ $lookup: {from: "enrollmets", localField:"_id", foreignField: "course_id", as: "enrollments" } }, { $unwind: "$enrollmets" }, { $lookup: { from:"students", localField: "enrollmets.student_id", foreignField: "_id", as: "student" } }, { $unwind:"$student" }, { $group: { _id: "$faculty_id", students: { $addToSet: "$student.name" },average_marks_per_student: { $avg: "$enrollmets.marks" } } }, { $lookup: { from: "faculty",localField: "_id", foreignField: "_id", as: "faculty" } }, { $unwind: "$faculty" }, { $project: { _id: 0,faculty_name: "$faculty.name", students: 1, average_marks_average_marks_per_student: { $round:["$average_marks_per_student", 2] } } }])

```
project_noSQL> db.faculty.aggregate([{ $lookup : { from : "courses_full"  , localField : "courses" , foreignField : "_id" , as : "course_info"}} , { $projec
t : { _id : 0 , faculty_name : "$name" , course_titles :"$course_info.title"} } ])  //PRIYA MAURYA roll no 1240258336
[
  { faculty_name: 'Alexis Stone', course_titles: [] },
  { faculty_name: 'Brooke Dorsey', course_titles: [] },
  { faculty_name: 'Kevin Booth', course_titles: [] },
  { faculty_name: 'Eduardo Mills', course_titles: [] },
  { faculty_name: 'Kevin Horton', course_titles: [] },
  { faculty_name: 'Michele Hines', course_titles: [] },
  { faculty_name: 'Joshua Wright', course_titles: [] },
  { faculty_name: 'Roberto Thompson', course_titles: [] },
  { faculty_name: 'Julie Elliott', course_titles: [] },
  { faculty_name: 'Meghan Watson', course_titles: [] },
  { faculty_name: 'Christian Campbell', course_titles: [] },
  { faculty_name: 'Stephen Galvan', course_titles: [] },
  { faculty_name: 'Traci Edwards', course_titles: [] },
  { faculty_name: 'Charles Cunningham', course_titles: [] },
  { faculty_name: 'Kevin Murphy', course_titles: [] },
  { faculty_name: 'Justin House', course_titles: [] },
  { faculty_name: 'Maurice Farmer', course_titles: [] },
  { faculty_name: 'Rhonda George', course_titles: [] },
  { faculty_name: 'Shelly Sawyer', course_titles: [] },
  { faculty_name: 'Paul Long', course_titles: [] }
]
```

Insight : Combines $group, $project , $lookup.

Q.14 : Show the most popular activity type (e.g., Hackathon, Seminar, etc.) by number of student participants.

Solution : db.Activities.aggregate([{ $group: { _id: "$type", participants: {$addToSet: "$student_id" } } }, { $project: { _id: 0, activity_type: "$_id", number_of_participants: {$size: "$participants" } } }, { $sort: { number_of_participants: -1 } }, { $limit: 1 }])

```
project_noSQL> db.Activities.aggregate([ { $group: { _id: "$type", participants: {$addToSet: "$student_id" } } }, { $project: { _id: 0, activity_type: "$_id
", number_of_participants: {$size: "$participants" } } }, { $sort: { number_of_participants: -1 } }, { $limit: 1 } ]) // priya maurya 1240258336
[ { activity_type: 'Hackathon', number_of_participants: 29 } ]
project_noSQL>
```

Insight : Combines $group, $sort, $limit,$project , $size .