// Inserting data into the professor collection

db.professor.insertMany([

{ profid: 1, name: "Professor A", age: 40, rank\_val: 1, research: "AI" },

{ profid: 2, name: "Professor B", age: 45, rank\_val: 2, research: "Machine Learning" },

{ profid: 3, name: "Professor C", age: 38, rank\_val: 3, research: "Data Science" }

])

// Inserting data into the projects collection

db.projects.insertMany([

{ pno: 1, sponsor: "UGC", sdate: ISODate("2006-01-01"), edate: ISODate("2008-12-31"), budget: 120000, p\_investigator: 1 },

{ pno: 2, sponsor: "AICTE", sdate: ISODate("2008-05-01"), edate: ISODate("2011-04-30"), budget: 150000, p\_investigator: 2 },

{ pno: 3, sponsor: "UGC", sdate: ISODate("2007-03-01"), edate: ISODate("2010-02-28"), budget: 90000, p\_investigator: 3 },

{ pno: 4, sponsor: "UGC", sdate: ISODate("2023-01-01"), edate: ISODate("2023-12-31"), budget: 150000, p\_investigator: 1 },

{ pno: 5, sponsor: "AICTE", sdate: ISODate("2023-06-01"), edate: ISODate("2024-05-31"), budget: 120000, p\_investigator: 2 },

{ pno: 6, sponsor: "UGC", sdate: ISODate("2023-08-01"), edate: ISODate("2024-07-31"), budget: 120000, p\_investigator: 1 },

{ pno: 7, sponsor: "AICTE", sdate: ISODate("2023-09-01"), edate: ISODate("2024-08-31"), budget: 150000, p\_investigator: 2 },

{ pno: 8, sponsor: "UGC", sdate: ISODate("2023-10-01"), edate: ISODate("2024-09-30"), budget: 90000, p\_investigator: 3 }

])

// Inserting data into the sworkson collection

db.sworkson.insertMany([

{ usn: 1001, pno: 1 },

{ usn: 1002, pno: 2 },

{ usn: 1003, pno: 3 }

])

// Inserting data into the pworkson collection

db.pworkson.insertMany([

{ profid: 1, pno: 1 },

{ profid: 3, pno: 1 },

{ profid: 2, pno: 2 },

{ profid: 3, pno: 2 },

{ profid: 3, pno: 3 },

{ profid: 1, pno: 4 },

{ profid: 3, pno: 4 },

{ profid: 2, pno: 5 },

{ profid: 3, pno: 5 },

{ profid: 1, pno: 6 },

{ profid: 3, pno: 6 },

{ profid: 2, pno: 7 },

{ profid: 3, pno: 7 },

{ profid: 1, pno: 8 },

{ profid: 2, pno: 8 },

{ profid: 3, pno: 8 }

])

// Inserting data into the student collection

db.student.insertMany([

{ usn: 1001, sname: "Student X", age: 25, degree: "MCA", profid: 1 },

{ usn: 1002, sname: "Student Y", age: 24, degree: "MPhil", profid: 2 },

{ usn: 1003, sname: "Student Z", age: 26, degree: "BE", profid: 3 }

])

// Professors without ongoing projects of more than 1 lakh

db.professor.aggregate([

{

$lookup: {

from: "pworkson",

localField: "profid",

foreignField: "profid",

as: "pworkson"

}

},

{

$lookup: {

from: "projects",

localField: "pworkson.pno",

foreignField: "pno",

as: "projects"

}

},

{

$match: {

$or: [

{ projects: { $size: 0 } },

{ "projects.budget": { $lte: 100000 } },

{ "projects.edate": { $lt: new Date() } }

]

}

},

{

$project: {

\_id: 0,

name: 1

}

}

])

// Graduate students, their professors, and project sponsors

db.student.aggregate([

{

$lookup: {

from: "professor",

localField: "profid",

foreignField: "profid",

as: "professor"

}

},

{

$lookup: {

from: "projects",

localField: "profid",

foreignField: "p\_investigator",

as: "projects"

}

},

{

$unwind: "$professor"

},

{

$project: {

\_id: 0,

student\_name: "$sname",

professor\_name: "$professor.name",

sponsor: "$projects.sponsor"

}

}

])

// Professors and sum of budgets of projects started after 2005 but ended in 2010

db.professor.aggregate([

{

$lookup: {

from: "projects",

localField: "profid",

foreignField: "p\_investigator",

as: "projects"

}

},

{

$unwind: "$projects"

},

{

$match: {

"projects.sdate": { $gte: new Date("2005-01-01") },

"projects.edate": { $lte: new Date("2010-04-10") }

}

},

{

$group: {

\_id: "$name",

totalBudget: { $sum: "$projects.budget" }

}

}

])

// Professors with total project worth greater than average project budget

db.professor.aggregate([

{

$lookup: {

from: "projects",

localField: "profid",

foreignField: "p\_investigator",

as: "projects"

}

},

{

$unwind: "$projects"

},

{

$group: {

\_id: "$name",

totalBudget: { $sum: "$projects.budget" }

}

},

{

$lookup: {

from: "projects",

let: { totalBudget: "$totalBudget" },

pipeline: [

{

$group: {

\_id: null,

avgBudget: { $avg: "$budget" }

}

}

],

as: "averageBudget"

}

},

{

$match: {

$expr: { $gt: ["$totalBudget", { $arrayElemAt: ["$averageBudget.avgBudget", 0] }] }

}

},

{

$project: {

\_id: 0,

name: "$\_id"

}

}

])

// Professors working on all projects

var allProjectsCount = db.projects.distinct("pno").length;

db.professor.aggregate([

{

$lookup: {

from: "pworkson",

localField: "profid",

foreignField: "profid",

as: "pworkson"

}

},

{

$lookup: {

from: "projects",

localField: "pworkson.pno",

foreignField: "pno",

as: "projects"

}

},

{

$addFields: {

projectCount: { $size: "$projects" }

}

},

{

$match: {

projectCount: allProjectsCount

}

},

{

$project: {

\_id: 0,

name: 1

}

}

])