QUESTION SET-10

**QUESTION-1: Jobs Collection**

**a) Retrieve all jobs that require either "Python" or "Java" as one of the skills.**

db.Jobs.find({ skillsRequired: { $in: ["Python", "Java"] } })

**b) Find jobs where the number of applicants is less than 50.**

db.Jobs.find({ "applicants": { $size: { $lt: 50 } } })

**c) Identify jobs posted by "Tech Innovators" with a salary range greater than $80,000.**

db.Jobs.find({

company: "Tech Innovators",

"details.salary": { $gt: 80000 }

})

**d) Sort jobs in ascending order of their posting date and skip the first 5 records.**

db.Jobs.find()

.sort({ "details.postingDate": 1 })

.skip(5)

**e) Group jobs by their company and count the total number of jobs posted per company.**

db.Jobs.aggregate([

{ $group: { \_id: "$company", totalJobs: { $sum: 1 } } }

])

**QUESTION-2: Candidates Collection**

**a) Retrieve candidates who have applied to more than 3 jobs but do not have "Data Science" as a qualification.**

db.Candidates.find({

appliedJobs: { $size: { $gt: 3 } },

qualifications: { $ne: "Data Science" }

})

**b) Find candidates who have applied to both "Software Engineer" and "Data Analyst" positions.**

db.Candidates.find({ appliedJobs: { $all: ["Software Engineer", "Data Analyst"] } })

**c) Display names and contact details of candidates with more than 5 years of experience.**

db.Candidates.find(

{ experience: { $gt: 5 } },

{ \_id: 0, name: 1, contact: 1 }

)

**d) Create a sparse index on the experience field to optimize queries filtering by years of experience.**

db.Candidates.createIndex(

{ experience: 1 },

{ sparse: true }

)

**e) Aggregate to project candidate’s name, total experience, number of jobs applied to, and filter candidates with fewer than 2 applications.**

db.Candidates.aggregate([

{ $project: { name: 1, experience: 1, totalJobsApplied: { $size: "$appliedJobs" } } },

{ $match: { totalJobsApplied: { $gte: 2 } } }

])