**Creation of database**

use guvi

**Creation and insertion of the collection “user”:**

db.createCollection(“users”)

db.users.insertMany([ { name: "Alice", email: "alice@example.com", batch: "Zen2024", mentor\_id: "mentor\_001" }, { name: "Bob", email: "bob@example.com", batch: "Zen2024", mentor\_id: "mentor\_001" }, { name: "Charlie", email: "charlie@example.com", batch: "Zen2024", mentor\_id: "mentor\_002" }, { name: "David", email: "david@example.com", batch: "Zen2024", mentor\_id: "mentor\_002" } ]);

**Creation and insertion of the collection “codekata”**

**db.createCollection(“codekata”)**

db.codekata.insertMany([ { user\_name: "Alice", problems\_solved: 50 }, { user\_name: "Bob", problems\_solved: 35 }, { user\_name: "Charlie", problems\_solved: 60 }, { user\_name: "David", problems\_solved: 20 } ]);

**Creation and insertion of the collection “attendance”**

**db.createCollection(“attendance”)**

db.attendance.insertMany([ { user\_name: "Alice", date: new Date("2020-10-16"), status: "Present" }, { user\_name: "Bob", date: new Date("2020-10-17"), status: "Absent" }, { user\_name: "Charlie", date: new Date("2020-10-18"), status: "Present" }, { user\_name: "David", date: new Date("2020-10-19"), status: "Absent" } ]);

**Creation and insertion of the collection “topics”**

**db.createCollection(“topics”)**

db.topics.insertMany([ { topic\_name: "Introduction to MongoDB", date: new Date("2020-10-05") }, { topic\_name: "Aggregation Framework", date: new Date("2020-10-10") }, { topic\_name: "Indexing in MongoDB", date: new Date("2020-10-15") } ]);

**Creation and insertion of the collection “tasks”**

**db.createCollection(“tasks”)**

db.tasks.insertMany([ { task\_name: "CRUD Operations", topic\_name: "Introduction to MongoDB", date: new Date("2020-10-06"), submitted\_by: ["Alice", "Charlie"] }, { task\_name: "Pipeline Stages", topic\_name: "Aggregation Framework", date: new Date("2020-10-12"), submitted\_by: ["Bob", "David"] }, { task\_name: "Indexing Examples", topic\_name: "Indexing in MongoDB", date: new Date("2020-10-16"), submitted\_by: ["Alice"] } ]);

**Creation and insertion of the collection “company\_drives”**

**db.createCollection(“company\_drives”)**

db.company\_drives.insertMany([ { company\_name: "TechCorp", date: new Date("2020-10-20"), students\_attended: ["Alice", "Charlie"] }, { company\_name: "InnoTech", date: new Date("2020-10-25"), students\_attended: ["Bob", "David"] } ]);

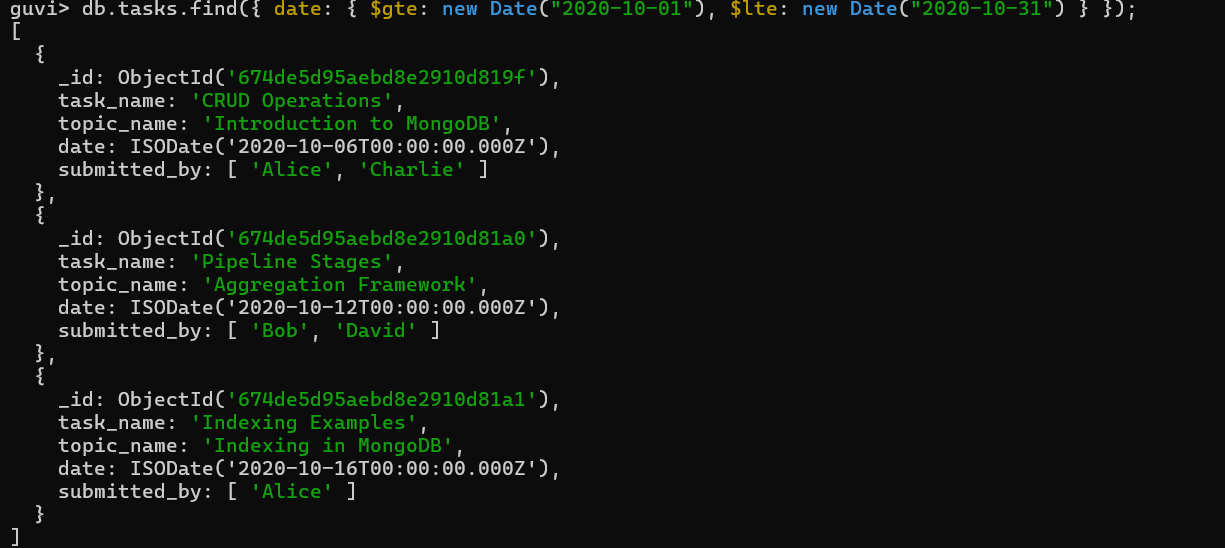
**Creation and insertion of the collection “mentors”**

**db.createCollection(“mentors”)**

db.mentors.insertMany([ { mentor\_id: "mentor\_001", mentor\_name: "John Doe", mentees: ["Alice", "Bob"] }, { mentor\_id: "mentor\_002", mentor\_name: "Jane Smith", mentees: ["Charlie", "David"] } ]);

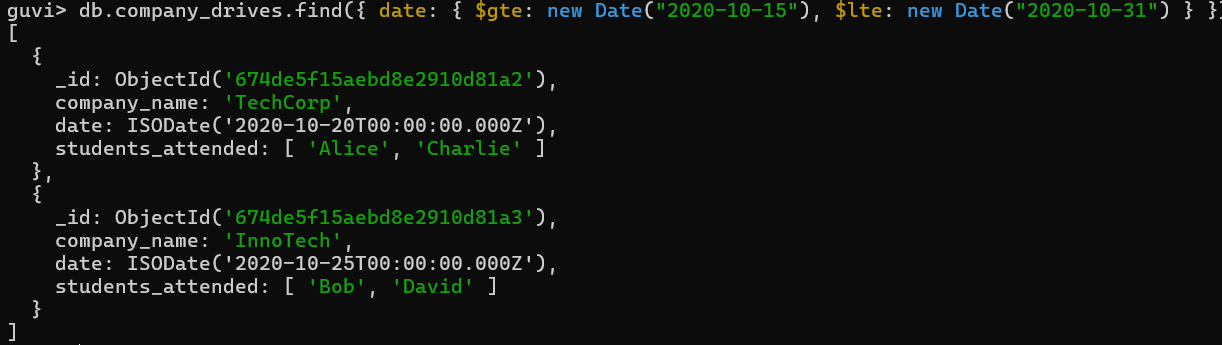
**1. Find topics and tasks taught in October 2020.**

db.tasks.find({ date: { $gte: new Date("2020-10-01"), $lte: new Date("2020-10-31") } });



**2. Find all company drives between 15-Oct-2020 and 31-Oct-2020.**

db.company\_drives.find({ date: { $gte: new Date("2020-10-15"), $lte: new Date("2020-10-31") } });



**3. Find all company drives and the students who appeared for placement.**

db.company\_drives.aggregate([

{

$lookup: {

from: "users",

localField: "students\_attended",

foreignField: "name",

as: "students\_details"

}

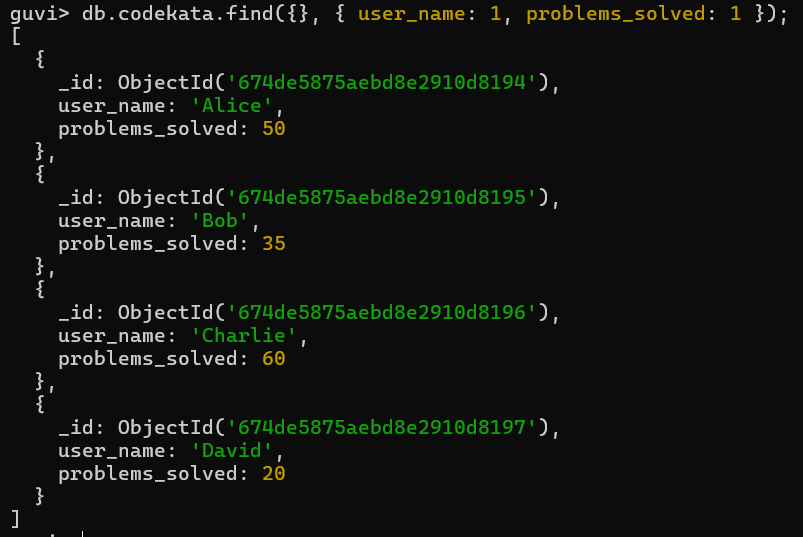
}

]);

****

**4. Find the number of problems solved by each user in CodeKata.**

db.codekata.find({}, { user\_name: 1, problems\_solved: 1 });



**5. Find all mentors with more than 15 mentees.**

**Adding more students to a mentor so that the above query will get results**

db.users.insertMany([ { name: "Eve", email: "eve@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" }, { name: "Frank", email: "frank@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" }, { name: "Grace", email: "grace@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" }, { name: "Hannah", email: "hannah@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" }, { name: "Ian", email: "ian@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" }, { name: "Jack", email: "jack@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" }, { name: "Karen", email: "karen@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" }, { name: "Leo", email: "leo@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" }, { name: "Mona", email: "mona@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" }, { name: "Nathan", email: "nathan@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" }, { name: "Olivia", email: "olivia@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" }, { name: "Paul", email: "paul@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" }, { name: "Quincy", email: "quincy@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" }, { name: "Rachel", email: "rachel@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" }, { name: "Steve", email: "steve@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" }, { name: "Tina", email: "tina@example.com", batch: "Zen2024", mentor\_id: "mentor\_003" } ]);

db.mentors.insertOne({ mentor\_id: "mentor\_003", mentor\_name: "Michael Brown", mentees: [ "Eve", "Frank", "Grace", "Hannah", "Ian", "Jack", "Karen", "Leo", "Mona", "Nathan", "Olivia", "Paul", "Quincy", "Rachel", "Steve", "Tina" ] });

db.mentors.find({ $expr: { $gt: [{ $size: "$mentees" }, 15] } });



**6. Find the number of users absent and tasks not submitted between 15-Oct-2020 and 31-Oct-2020.**

**Inserting additional data inorder to get the result**

db.attendance.insertMany([

{ user\_name: "Alice", date: new Date("2020-10-16"), status: "Absent" },

{ user\_name: "Bob", date: new Date("2020-10-17"), status: "Absent" },

{ user\_name: "Charlie", date: new Date("2020-10-18"), status: "Present" },

{ user\_name: "David", date: new Date("2020-10-19"), status: "Absent" },

{ user\_name: "Eve", date: new Date("2020-10-22"), status: "Absent" },

{ user\_name: "Frank", date: new Date("2020-10-23"), status: "Absent" },

{ user\_name: "Grace", date: new Date("2020-10-24"), status: "Absent" }

]);

db.tasks.insertMany([

{ task\_name: "CRUD Operations", topic\_name: "Introduction to MongoDB", date: new Date("2020-10-06"), submitted\_by: ["Alice", "Charlie"] },

{ task\_name: "Pipeline Stages", topic\_name: "Aggregation Framework", date: new Date("2020-10-12"), submitted\_by: ["Bob", "David"] },

{ task\_name: "Indexing Examples", topic\_name: "Indexing in MongoDB", date: new Date("2020-10-16"), submitted\_by: ["Alice", "Charlie"] },

{ task\_name: "Advanced Aggregation", topic\_name: "Aggregation Framework", date: new Date("2020-10-20"), submitted\_by: ["Charlie"] },

{ task\_name: "MongoDB Indexing", topic\_name: "Indexing in MongoDB", date: new Date("2020-10-22"), submitted\_by: [] },

{ task\_name: "Aggregation Optimization", topic\_name: "Aggregation Framework", date: new Date("2020-10-23"), submitted\_by: [] },

{ task\_name: "Aggregation Pipeline", topic\_name: "Aggregation Framework", date: new Date("2020-10-24"), submitted\_by: [] }

]);

db.attendance.aggregate([

{

$match: {

date: { $gte: new Date("2020-10-15"), $lte: new Date("2020-10-31") },

status: "Absent"

}

},

{

$lookup: {

from: "tasks",

localField: "user\_name",

foreignField: "submitted\_by",

as: "tasks\_submitted"

}

},

{

$match: { tasks\_submitted: { $eq: [] } }

}

]);

