**users**

* + user\_id: ObjectId
  + name: String
  + email: String
  + mentor\_id: ObjectId (Reference to mentors)
  + batch: String

**codekata**

* + codekata\_id: ObjectId
  + user\_id: ObjectId (Reference to users)
  + problems\_solved: Number
  + date\_solved: Date

**attendance**

* + attendance\_id: ObjectId
  + user\_id: ObjectId (Reference to users)
  + date: Date
  + status: String (e.g., "Present", "Absent")

**topics**

* + topic\_id: ObjectId
  + topic\_name: String
  + date: Date

**tasks**

* + task\_id: ObjectId
  + topic\_id: ObjectId (Reference to topics)
  + task\_name: String
  + date\_assigned: Date
  + date\_submitted: Date
  + user\_id: ObjectId (Reference to users)

**company\_drives**

* + drive\_id: ObjectId
  + company\_name: String
  + date: Date
  + students\_appeared: Array of ObjectId (Reference to users)

**mentors**

* + mentor\_id: ObjectId
  + name: String
  + mentees: Array of ObjectId (Reference to users)

**Queries**

**1.Find all the topics and tasks which are taught in the month of October**:

db.topics.aggregate([

{

$lookup: {

from: "tasks",

localField: "topic\_id",

foreignField: "topic\_id",

as: "tasks"

}

},

{

$match: {

$expr: { $eq: [{ $month: "$date" }, 10] }

}

}

])

**2.Find all the company drives which appeared between 15 Oct 2020 and 31 Oct 2020**:

db.company\_drives.find({

date: {

$gte: ISODate("2020-10-15T00:00:00Z"),

$lte: ISODate("2020-10-31T23:59:59Z")

}

})

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

db.company\_drives.aggregate([

{

$lookup: {

from: "users",

localField: "students\_appeared",

foreignField: "user\_id",

as: "students"

}

}

])

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

db.codekata.aggregate([

{ $group: { \_id: "$user\_id", total\_problems\_solved: { $sum: "$problems\_solved" } } }

])

**5.Find all the mentors who have mentees count more than 15**:

db.mentors.find({

$expr: { $gt: [{ $size: "$mentees" }, 15] }

})

**6.Find the number of users who are absent and task is not submitted between 15 Oct 2020 and 31 Oct 2020**:

db.attendance.aggregate([

{

$match: {

date: {

$gte: ISODate("2020-10-15T00:00:00Z"),

$lte: ISODate("2020-10-31T23:59:59Z")

},

status: "Absent"

}

},

{

$lookup: {

from: "tasks",

localField: "user\_id",

foreignField: "user\_id",

as: "tasks"

}

},

{

$match: {

"tasks.date\_submitted": { $exists: false }

}

},

{

$group: {

\_id: null,

total\_absent\_and\_not\_submitted: { $sum: 1 }

}

}

])